Universität Leipzig
Institut für Informatik

Diplomarbeit

Design und Implementierung eines Algorithmus zum maschinellen Lernen der Flexion eines Korpus deutscher Sprache

Vorgelegt von Julian Moritz

Betreuer Prof. Dr. Uwe Quasthoff
Abteilung Automatische Sprachverarbeitung

Leipzig – Dezember 2009

ZUSAMMENFASSUNG

Die vorliegende Arbeit beschreibt das Design und die Implementierung eines Algorithmus zur Flexion. Es wird am Beispiel des Deutschen eine konkrete Implementierung entwickelt. Hierfür findet zunächst eine ausführliche Analyse der Flexion des Deutschen statt, bevor ein Verfahren erarbeitet wird, das sprachunabhängig ist und somit prinzipiell auf andere Sprachen übertragen werden kann.

Die tatsächliche Machbarkeit des Verfahrens wird anhand von Beispielen nachgewiesen. Die hohe Komplexität der Aufgabe führt allerdings dazu, dass es in der Praxis zu Abstrichen bei der Qualität der flektierten Wortformen kommt. Dies ist insbesondere deswegen der Fall, da das entwickelte System auch ihm unbekannte Grundformen flektiert.

DER WERWOLF

Ein Werwolf eines Nachts entwich
von Weib und Kind und sich begab
an eines Dorfschullehrers Grab
und bat ihn: „Bitte, beuge mich!“

Der Dorfschulmeister stieg hinauf
auf seines Blechschilds Messingknauf
und sprach zum Wolf, der seine Pfoten
geduldig kreuzte vor dem Toten:

„Der Werwolf“ – sprach der gute Mann,
„des Weswolfs, Genitiv sodann,
dem Wemwolf, Dativ, wie man’s nennt,
den Wenwolf, – damit hat’s ein End.“

Dem Werwolf schmeichelten die Fälle,
er rollte seine Augenbälle.
Indessen, bat er, füge doch
zur Einzahl auch die Mehrzahl noch!

Der Dorfschulmeister aber musste
gestehn, dass er von ihr nichts wusste.
Zwar Wölfe gäb’s in grosser Schar,
doch „Wer“ gäb’s nur im Singular.

Der Wolf erhob sich tränenblind –
er hatte ja doch Weib und Kind!
Doch da er kein Gelehrter eben,
so schied er dankend und ergeben.

Christian Morgenstern

INHALTSVERZEICHNIS

TABELLENVERZEICHNIS

ABBILDUNGSVERZEICHNIS

INHALT

1

EINFÜHRUNG

1.1 VORWORT

In den letzten Jahren hat sich der Wissenschaft durch das Internet eine unüberschaubare Menge an geschriebener Sprache erschlossen. Technische und wissenschaftliche Entwicklungen machen es nunmehr möglich, aus dieser Quelle Wörter zu extrahieren, die den deutschen Wortschatz abbilden.

Doch diese Listen weisen Lücken auf, denn selten und nicht vollständig flektiert vorkommende Wörter fehlen, zudem finden sich Zeichenketten in den Listen, die nicht als Wörter gelten. Das Auffinden und Löschen ist mühselig, meist vom Zufall abhängig und kann nie sicher als vollständig abgeschlossen betrachtet werden[1].

Diese Arbeit soll dem Abhilfe schaffen: anstatt möglichst viel Text zu sammeln und daraus die Wörter zu extrahieren und ihren Grundformen zuzuordnen, und währenddessen auf Vollständigkeit zu hoffen, werden für die bekannten Grundformen systematisch alle Wörter gebildet.

1.1.1 *Motivation*

Die Einsatzmöglichkeiten für Grundformen und ihre flektierten Formen sind vielfältig:

- Durch manuelle Wortvervollständigung auf Endgeräten mit eingeschränkter Tastatur erleichtern Wortlisten die Eingabe um so besser, je umfangreicher sie sind.
- Für das Erlernen einer fremden Sprache kann es für den Lernenden nützlich sein, sich alle flektierten Formen eines Wortes anzeigen zu lassen.

1 Es sei denn, die Wortliste ist hinreichend klein, dass sie manuell überprüft werde kann.

- Für die automatische Rechtschreibkorrektur ist der Ansatz denkbar, Wörter nur dann zu korrigieren, wenn sie nicht in einer Wortliste auftauchen, die nach den Normen der Rechtschreibung gebildet wurde.

- Ferner helfen Wortlisten bei der automatischen Wortvervollständigung, etwa bei Texten, die nur bruchstückhaft vorhanden sind, sowie bei der Texterkennung (*Optical Character Recognition*).

Sicherlich liegen auch große Wörterbücher mit Angaben zur Flexion bereits digital vor, allerdings ist eine manuelle Pflege notwendig, da jederzeit neue Wörter im Sprachgebrauch auftauchen.[2] Hier liefert ein Wörterbuch ohne passenden Eintrag nichts zurück.

Zudem können große Wörterbücher wie der Duden aus urheberrechtlichen Gründen nicht frei für vielfältige Zwecke verwendet werden; der Nutzen hält sich also durchaus in Grenzen.

1.2 ZIEL

Ziel dieser Arbeit ist der Entwurf und die Implementierung eines Algorithmus, der nach einer Trainingsphase bestimmt, wie Grundformen (der deutschen Sprache) flektiert werden. Die Implementierung trägt den Namen FLEXI[3], der im Folgenden immer genau dann verwendet wird, wenn über die Implementierung des Algorithmus aus Arbeit gesprochen wird, vor allem, um Verwechslungen mit anderen Implementierungen zu vermeiden.

Hervorzuheben ist, dass auch unbekannte Grundformen, also solche, die nicht zu den Trainingsdaten gehören, möglichst korrekt flektiert werden sollen.

Hierfür sind zwei Teilaufgaben zu lösen:

1. Zunächst ist eine Trainingsmenge zu erstellen, die Daten darüber enthält, wie Grundformen flektiert

2 Laut [Qua07] sind von 2000 bis 2007 2284 neue Wörter in die deutsche Alltagssprache aufgenommen worden; dies kann dadurch geschehen, dass die Wörter völlig neu entstanden sind, bisher nur Spezialisten bekannt waren, an bestimmte Ereignisse gebunden sind, mit technischen Entwicklungen zu tun haben, plötzlich häufiger verwendet wurden oder eine ganz neue Bedeutung erlangt haben.

3 Der Name ist inspiriert von dem lateinischen *flexi*, übersetzt *ich habe gebeugt*.

werden und aus der hervorgeht, was dazu führt, wie eine bestimmte Grundform flektiert wird.[4]

2. Daraufhin ist maschinell zu erlernen, welche Veränderungen wodurch bestimmt werden, so dass diese Veränderungen auf andere Grundformen übertragbar sind. Es nützt nämlich einerseits wenig, wenn klar ist, dass *Fisch* genau so flektiert wird wie *Tisch*, der Algorithmus zwar die eine, doch nicht die andere Grundform flektieren kann. Wünschenswert ist z.B., dass die Veränderung von *Blatt* zu *Blätter* auf *Rad* anwendbar ist, so dass *Räder* gebildet werden kann, obwohl der umlautende Vokal an anderer Stelle steht. Andererseits sollte der Algorithmus natürlich bestimmen können, dass *Fisch* genau so flektiert wird wie *Tisch*, falls sich die eine, aber nicht die andere Grundform in der Trainingsmenge befindet.

Der zweite Schritt sollte möglichst sprachunabhängig implementiert werden.[5]

Es ist zu erwähnen, dass unregelmäßig flektierende Wörter (wie z.B. *sein*) in der Trainingsmenge enthalten sein müssen, da deren Flexion nicht abgeleitet werden kann.

1.2.1 *Vorgehen*

In Kapitel 2 wird die Flexion des Deutschen beschrieben; so wird klar, welche Veränderungen erlernt werden müssen und wodurch diese bestimmt werden. Kapitel 3 widmet sich der bereits bestehenden Software, die Grundformen des Deutschen flektieren kann. Es soll ein Überblick geschaffen werden, was bereits implementiert wurde. Zudem können so die Fehler anderer vermieden, die Erfolge genutzt werden. Der in dieser Arbeit entwickelte Algorithmus wird in Kapitel 4 beschrieben, so dass dessen Funktionsweise nachvollziehbar wird. Abschließend findet eine Evaluation in Kapitel 5 statt.

4 Es wird in Kapitel 2 beschrieben, wodurch Flexion bestimmt wird.

5 Es ist in der vorgegebenen Zeit sicher nicht möglich, den Algorithmus auf alle schriftlich dokumentierten, flektierenden Sprachen hin zu testen. Es wird lediglich versucht, die sprachlichen Eigenheiten des Deutschen so zu verallgemeinern, dass eine Übertragung auf andere Sprachen möglich bleibt.

2

FLEXION IM DEUTSCHEN

2.1 EINLEITUNG

In dieser Arbeit geht es um die Generierung von Worten mittels Flexion. Doch was ist eigentlich ein Wort? Und wie genau wird im Deutschen flektiert? Im Folgenden werden zunächst einige linguistische Grundbegriffe erläutert, bevor ausführlich darauf eingegangen wird, wodurch die Flexion bestimmt wird und wie sich die Grundformen durch sie verändern.

2.1.1 *Grundbegriffe*

Eine allgemeingültige, eindeutige Definition des Begriffes **Wort** existiert nicht. Es wird je nach Kontext und Verständnis unterschiedlich und teilweise widersprüchlich charakterisiert. Auf der orthographisch-graphemischer Ebene werden Wörter „durch Leerstellen im Schriftbild isoliert“ (vgl. [Bus90]). Auf der morphologischen Ebene jedoch sind Wörter „als Grundeinheiten von grammatischen Paradigmen wie Flexion gekennzeichnet und zu unterscheiden von den morphologisch charakterisierten Wortformen“ (vgl. ebenda). So wird im Folgenden der Begriff des Wortes im Sinne der morphologischen Wortform, der Begriff der Grundform im Sinne des morphologischen Wortes verwendet.

Als **Morphem** wird die kleinste, grammatikalisch bedeutungstragende sprachliche Einheit bezeichnet. Morpheme sind die Grundbausteine der Wörter, besitzen eine selbstständige Bedeutung und haben eine eindeutig feststellbare Form und Funktion (vgl. [Rö06, S. 27]).

Als **Morph** werden die unterschiedlichen Ausprägungen eines Morphems bezeichnet. Das Morphem für das Merkmal *Plural* von Substantiven hat z.B. die Morphe *-s* (vgl. *Auto*, im Plural *Auto-s*) und *-e* (z.B. *Hund*, im Plural *Hund-*

e). Dass ein Morphem in Varianten auftreten kann, nennt man **Allomorphie**.

Je nach ihrer Funktion im Satz lassen sich Wörter in **Wortarten** einteilen. Je nach Wortart muss ein Wort bestimmte morphosyntaktische Kategorien ausdrücken und ihm müssen bestimmte Morpheme für die Merkmale zugeordnet werden können. So kann das Substantiv *Haus* nicht das Morphem für die 3. Person Singular Präsens Aktiv *t* annehmen (anders als z.B. *glauben*: *glaub-**t***).

2.1.2 *Flexion*

Nach [MA05] ist **Flexion** der Prozess, durch den die Form eines Wortes seiner Position bzw. Funktion im Satz angepasst wird. Genauer spricht man von der Bildung verschiedener Wortformen zum Ausdruck bestimmter morphosyntaktischer Merkmale.

In der generativen Linguistik gibt es verschiedene Ansätze, die Wortbildung durch Flexion mit einem Algorithmus abzubilden (z.B. morphembasierte, wortbasierte bzw. realisierungsbasierte, vgl. [Dra04, S. 194]). Für FLEXI wird ein pragmatischer Ansatz[1] gewählt: Die Flexion wird als so genannte *Black Box* aufgefasst, durch die aus einer Grundform die flektierten Formen gebildet werden können. Dafür sind – neben der Morphologie – folgende Ebenen von Bedeutung:

SYNTAX Die Syntax ist die Lehre vom Satzbau. Eine Änderung der syntaktischen Struktur des Satzes kann eine Änderung der Wortform zur Folge haben (z.B. *. . . , weil ich **weggehe**.* und *Ich **gehe weg**.*).

SEMANTIK Inhalt der Semantik ist die Bedeutung. Ist ein Wort ein Homograph (also ein Wort aus einer Gruppe von Wörtern, die zwar die gleiche Schreibweise, aber unterschiedliche Bedeutungen haben), so kann es je nach Bedeutung unterschiedlich flektiert werden (vgl. *hängen: Ich **hängte** das Bild auf.* oder *Ich **hing** fest.*).

1 Pragmatisch deshalb, weil nicht verschiedene Komponenten integriert werden. So wird z.B. nicht unterschieden, ob eine Allomorphie phonologisch oder morphologisch bedingt ist, z.B. die phonologische Allomorphie in der 2. Person Singular Präsenz der Verben auf *s* (wie in *hass**t***), auf *t* (wie in *wat**est***) oder andere (wie in *hol**st***, *schau**st***), vgl. [Dra04, S. 200]. Es werden lediglich die Veränderungen an den Grundformen erfasst und wodurch diese bestimmt werden.

PHONOLOGIE Die Systematik bei der Verwendung von Lauten einer Sprache ist Gegenstand der Phonologie. Bestimmte Substantive haben nach [TV09] im Plural das Suffix *e* oder ε[2], abhängig lediglich davon, ob die letzte Silbe des Wortes betont ist oder nicht (z.B. *die Ausblicke* und *die Messer*).

ORTHOGRAPHIE Auch die Rechtschreibung kann bestimmend sein für die Form eines flektierten Wortes (z.B. *gießen* und *gossen*, in diesem Fall wird das *ß* nach dem langen Vokal *ie* zu *ss* nach dem kurzen Vokal *o*).

ETYMOLOGIE Die Etymologie ist der Zweig der Linguistik, der sich mit der Herkunft und Entwicklung der Wörter beschäftigt. Existieren zwei Wörter unterschiedlicher Herkunft, aber mit identischer Schreibweise, kann dies Grund zu einer unterschiedliche Flexion führen (z.B. *die Bank, die Banken* von dem italienischen *banca* und *die Bank, die Bänke* von dem althochdeutschen *banc*).

Die einzelnen **morphosyntaktischen Kategorien** (auch *Merkmalklassen*, nach [TV09]) mit ihren Merkmalen sind in Tabelle 2.1 auf Seite 9 aufgelistet.

Kategorie	Merkmale
Numerus	Singular, Plural
Kasus	Nominativ, Genitiv, Dativ, Akkusativ
Genus	Maskulinum, Femininum
Genus verbi	Aktiv, Passiv
Modus	Indikativ, Konjunktiv, Imperativ
Komparation	Positiv, Komparativ, Superlativ
Person	Erste, Zweite, Dritte
Tempus	Präsenz, Präteritum, Perfekt, Plusquamperfekt, Doppelperfekt, Doppelplusquamperfekt, Futur I, Futur II

TABELLE 2.1: Tabellarische Auflistung der morphosyntaktischen Kategorien und ihrer Merkmale im Deutschen

Mit diesen Merkmalen kann für jedes (flektierende) Wort dessen Grundform bestimmt werden. Die Grundformen

2 ε steht für die leere Zeichenkette.

(oder auch *Nennformen*) nach Wortarten sind wie folgt allgemein festgelegt:

SUBSTANTIV Nominativ, Singular

ADJEKTIV Nominativ, Singular, Maskulin, Positiv

VERB Infinitiv, Präsenz

ARTIKEL/PRONOMEN Nominativ, Singular, Maskulin

ADVERB Positiv

Ausnahmen sind z.B. Adjektive, die nur in gesteigerter Form auftreten, wie z.B. *allerschönster* und *weltbester* oder Substantive, die nur im Plural auftreten, wie z.B. *Eltern* und *Streitkräfte*.

2.2 WORTFORMEN NACH WORTARTEN

Der folgende Abschnitt bezieht sich hauptsächliche auf [TV09], [KRSSW07] und [fdS06].[3] Die beschriebenen Veränderungen beziehen sich nicht allein auf eine spezielle Ebene der Veränderung, sondern beschreiben, welche Veränderungen an der Grundform eines Wortes auftreten können, wenn sie in einem geschriebenen Text verwendet wird.

Zu den flektierenden Wortarten gehören im Allgemeinen Substantiv, Adjektiv, Verb, Artikel und Pronomen. Auch Adverbien sind vereinzelt komparierbar (z.B. *oft*, *öfter*), zählen allerdings zu den nicht-flektierenden Wortarten, die außerdem noch Präposition, Konjunktion und die Partikel beinhalten.

In den folgenden Ausführungen wird erläutert, wie sich die Grundformen verändern und durch welche Eigenschaften diese Veränderungen hervorgerufen werden.

3 Diese drei Werke finden aus folgendem Grund Verwendung: *Flexion* ([TV09]) erhebt den Anspruch, die erste Einführung in die Flexionsmorphologie zu sein. Es beschränkt sich hierbei jedoch auf den morphologischen Teil der Flexion, nicht auf die anderen Ebenen wie Semantik, Orthographie, etc. Es wird z.B. nicht erwähnt, dass Eigennamen im Genitiv in bestimmten Fällen ein Apostroph bekommen (vgl. *Aristoteles' Schriften* oder *Alice' neue Wohnung*). Ergänzend kommt daher das Regelwerk des Rats für deutsche Rechtschreibung ([fdS06]) hinzu, das Regeln für die deutsche Rechtschreibung festlegt. Der Duden ([KRSSW07]) wird hingegen genutzt, um die Flexion bestimmter Einzelfälle nachzuschlagen.

2.2.1 *Substantiv*

Bei der Deklination des Substantivs werden die Merkmale *Numerus* und *Kasus* ausgedrückt, jedes Substantiv besitzt auch ein unveränderliches Merkmal *Genus*.

Es gibt jedoch selten Substantive, die nicht vollständig dekliniert werden, so kommt z.B. *Leute* nur im Plural vor, *Verstand* nur im Singular.

NOMINATIV, SINGULAR Der Nominativ im Singular ist der unmarkierte Fall und die Grundform; Ausnahmen sind die Substantive, die nur im Plural stehen können: hier wird der Nominativ Plural verwendet.

GENITIV, SINGULAR Die Feminina markieren im Singular im Allgemeinen keinen Kasus. Bei den Maskulina und Neutra (somit Nicht-Feminina) gibt es drei unterschiedliche Möglichkeiten zur Genitiv-Bildung:

1. Ist die letzte Silbe eine Schwasilbe, so tritt das Suffix *s* auf.
2. Ist die letzte Silbe keine Schwasilbe und lautet die Grundform nicht auf *s* aus, so tritt das Suffix *[e]s*[4] auf.
3. Ist die letzte Silbe keine Schwasilbe und lautet die Grundform auf *s*, *x*, *ß* oder *z* aus, so tritt das Suffix *es* auf.

Ausnahmen sind die Eigennamen (vgl. [fdS06, S. 98]):

- Eigennamen, die auf *s*, *ß*, *z*, *x* oder *ce* enden, haben im Genitiv ein Apostroph, falls sie von einem Artikel, einem Possesivpronomen oder ähnlichem begleitet werden.
- Eigennamen (auch feminine), die nicht auf einen /s/-Laut enden, haben im Genitiv das Suffix *s*, insofern sie ohne Begleiter auftreten.
- Geografische Namen enthalten – insofern sie Neutrum oder Maskulinum – sind, auch das Genitiv-Suffix *s*, falls sie ohne Artikel stehen.

4 Der Buchstabe in den eckigen Klammern ist optional; er kann, muss aber nicht verwendet werden, was nicht von einem bestimmten Merkmal abhängig ist (vgl. *des Wolfs* und *des Wolfes*)

DATIV, SINGULAR Die Nicht-Feminina, die nicht auf eine Schwasilbe enden, bilden den Dativ mit dem Suffix *e*; dieses gilt jedoch als veraltet und ist optional. Infolge dessen wird das Dativ-Suffix mit *[e]* bezeichnet.

AKKUSATIV, SINGULAR Die Akkusativ-Form ist identisch mit der Nominativ-Form.

NOMINATIV, GENITIV, AKKUSATIV, PLURAL Die drei Fälle Nominativ, Genitiv und Akkusativ sind im Plural unmarkiert; es gibt 5 unterschiedliche Plural-Suffixe:

$\emptyset$ Nicht-Feminia, die auf eine Schwasilbe enden.

e Nicht-Feminia, die nicht auf eine Schwasilbe enden.

n Feminia, die auf eine Schwasilbe enden.

en Feminia, die nicht auf eine Schwasilbe enden.

s Mehrsilbige Substanive, die auf einen Vollvokal enden.

Desweiteren tritt der Effekt der Umlautung nach bestimmten Regeln auf:

- Substantive mit den Plural-Suffixen *n*, *en* oder *s* lauten nicht um (einzige Ausnahme: *die Werkstätten*).
- Ist der Stammvokal umlautfähig, lauten Wörter mit dem Plural-Suffix *er* immer um.
- Alle Feminia mit dem Plural-Suffix *e* lauten um, insofern dies möglich ist.
- Neutra ohne Pluralsuffix haben nie einen Umlaut (einzige Ausnahmen: *Klöster*, *Wässer*).
- Hat ein Neutrum das Plural-Suffix *e*, so lautet es ebenfalls nicht um (einzige Ausnahme: *Flöße*).
- Bei den Maskulina mit der Möglichkeit zur Umlautung und den beiden Plural-Suffixen *e* und $\emptyset$ gibt es solche, die nicht umlauten, als auch solche, die umlauten.

Zur letzten Umlautungs-Regel ist noch hinzuzufügen, dass bei den einsilbigen Substantiven die Verteilung etwa

50:50 beträgt. Im Gegensatz dazu lauten etwa 90 % der zweisilbigen Substantive um, die kein Plural-Suffix haben.

Das Plural-Suffix *s* tritt außerdem bei Eigennamen (z.B. *Peters*), Kurzwörtern (z.B. *Akkus*), Buchstabenwörtern (z.B. *PKWs*), Onomatopoetika (z.B. *Wauwaus*), Substantivierungen (z.B. *Warums*) und einer Reihe von Fremdwörtern (z.B. *Bars*) auf.

DATIV, PLURAL Der Dativ wird im Plural nur markiert, wenn der Plural mit dem Suffix *e* oder *er* gebildet wird; die Markierung erfolgt mit dem Suffix *n*.

2.2.2 *Adjektiv*

Durch die Flexion des typischen Adjektivs werden die Merkmale Genus, Numerus und Kasus zum Ausdruck gebracht. Jedes flektierende Adjektiv kann zudem stark (in Begleitung des definiten Artikels), schwach (ohne Artikel) und gemischt (in Begleitung von *ein*, *kein* und *mein*) auftreten. *In Begleitung* meint jedoch nicht, dass der Artikel direkt mit diesem Adjektiv auftreten muss, sondern er kann auch eine Aufzählung von Adjektiven begleiten, die sich dann alle nach diesem richten (z.B. *Neue, starke Männer braucht das Land.* und *Die neue**n**, starke**n** Männer braucht das Land.*). Ferner kann das (typische) Adjektiv kompariert werden.

Gebraucht werden kann ein Adjektiv auf drei Arten:

ATTRIBUTIV *Der schnelle Hund läuft.*

PRÄDIKATIV *Der Hund ist schnell.*

ADVERBIAL *Der Hund läuft schnell.*

Schnell finden sich jedoch Beispiele für Adjektive, die nicht flektieren (vgl. *super*, *rosa*, *Leipziger*) oder nicht steigerbar sind (z.B. *täglich*). Auch werden Adjektive, die einem Substantiv nachgestellt sind (vgl. *Sonne pur*) oder in bestimmten Phrasen vorkommen (vgl. *ruhig Blut*, *lieb Kind*, *ganz Leipzig*), nicht flektiert. Ferner lässt sich nicht jedes Adjektiv auf jede Art gebrauchen (z.B. *damalig*, *ständig*).

FLEXION Auf den ersten Blick werden alle Adjektive gleich gebeugt, es lässt sich allerdings folgendes festhalten:

- Bei den flektierenden Adjektiven, die auf *e* enden, werden die Suffixe ohne *e* verwendet (vgl. *ein müde**r** Mann*).

- Ist die letzte Silbe eines Adjektivs *en*, *el* oder *er* und wird sie betont, kann das *e* beim Anhängen eines Suffixes ausgelassen werden, siehe:

  EL spendabler Mann – spendabeler Mann

  ER saubrer Raum – sauberer Raum

  EN trockner Wein – trockener Wein

- Eine spezielle Ausnahme in der Flexion ist das Wort *hoch* (vgl. *der hohe Baum*), da hier das *c* in den flektierenden Formen nicht vorkommt.

Substantivierte Adjektive bezeichnen entweder Personen (vgl. *die Verlobte*, *ein Erwachsener*) oder Abstrakte Begriffe (vgl. *das Schöne*, *alles Gute*). Erstere werden wie herkömmliche Adjektive stark, schwach und gemischt – allerdings nur im Maskulinum und Femininum vor kommend – flektiert, letztere – ausschließlich im Neutrum vorkommend – nach *etwas*, *nichts* und *viel* stark, nach *alles* oder einem bestimmten Artikel schwach.

KOMPARATION Neben dem unflektierten Positiv – der unmarkierten Form des Adjektivs – lassen sich zwei weitere Formen bilden: Komparativ und Superlativ. Zunächst zum **Komparativ**: Er wird mit dem Suffix *er* gebildet, wobei hier die gleichen Umstände eine Rolle wie bei den Suffixen des Positivs (vgl. *Er ist müde**r** als sie.*).

Der **Superlativ** wird durch Anfügen von *est* oder *st* an die unflektierte Form des Positivs gebildet. Im unmarkierten Fall wird der Superlativ durch *st* gebildet, endet ein Adjektiv jedoch auf *d*, *s*, *sk*, *ß*, *t*, *x* oder *z* und enthält die letzte Stammsilbe einen Vollvokal, so wird *est* angefügt (einzige Ausnahme: *groß*). Endet ein Adjektiv auf *sch*, einen Diphtong oder einen betonten Vollvokal, lässt sich der Superlativ sowohl mit *st* als auch mit *est* bilden.

Umlautung findet in der Regel bei Komparativ und Superlativ nicht statt, jedoch gibt es Adjektive, die immer Umlauten (z.B. *alt*, *hart*, *gesund*), als auch solche, die umlauten können, aber nicht müssen (z.B. *bang*, *blass*).

Die fünf Ausnahmen bilden die Adjektive *hoch*, *nah*, *viel*, *gut* und *groß*.

ADJEKTIVE, DIE NICHT STEIGERBAR SIND Die Nicht-Komparierbarkeit vor einen semantischen Ursprung. Es können nicht gesteigert werden:

- Nur Attributiv verwendete Adjektive wie *heutig*, *damalig* oder *stündlich*, die also einen Zeitraum oder -punkt beschreiben, können nicht gesteigert werden.
- Partizipien in der Funktion von Adjektiven (z.B. *schreibend*, *fahrend*).
- Ordinalzahlwörter oder Adjektive, die Zahlwörter enthalten (vgl. **der drittere Platz*, **der vierfachere Vater*).
- Adjektive, deren Eigenschaft nicht in einem unterschiedlichen Maß vorliegen kann (z.B. *tot*, *rund*).
- Adjektive, die einen verstärkenden Anteil haben (z.B. *riesengroß*, *pudelwohl*).
- Adjektive, die einen verneinenden Anteil haben (z.B. *lustlos*, *unlustig*).

Trotzdem kommen diese Adjektive gesteigert vor, vor allem, wenn sie nicht in ihrer Ursprünglichen Form verwendet werden (vgl. *Und vielleicht ist der Opa jetzt lebendiger, als er es je war.* [Die09]).

Es gibt einige Sonderfälle, von denen kein Positiv und Komparativ gebildet werden kann (z.B. *weltbester*, *allerschönster*); dies hat ebenfalls semantische Gründe.

WORTGRUPPEN Da häufig der korrekte Gebrauch unklar ist, folgt ein kleiner Exkurs zu der Steigerung von Wortgruppen: In der geschriebenen Sprache finden sich Konstruktionen wie *die schwerwiegendensten Vorwürfe* und *die schwerstwiegenden Vorwürfe*; im Gesprochenen fälschlicherweise auch **schwerstwiegendsten*[5]. Dabei ist es eigentlich recht einfach: der Positiv von *schwerstwiegend* ist *schwer wiegend*, der von *schwerwiegendst schwerwiegend*. Konstruktionen wie **schwerstwiegendst* oder **nächstliegendst* werden fälschlicherweise ausnahmslos von Wortgruppen gebildet, bei denen nur das Erstglied gesteigert werden dürfte.

5 Wie ein Blick in das Internet zeigt, auch im Geschriebenen, z.B. [Bre09].

2.2.3 *Verb*

Durch die Konjugation des Verbs werden die Merkmale Person, Numerus, Tempus, Modus und Genus Verbi ausgedrückt. Will man beschreiben, wie ein bestimmtes Verb gebeugt wird, gibt man drei Formen an: den einfachen Infinitiv (die **Nennform**), 1. Person Präteritum aktiv und das Partizip II.

Anhand dieser Formen kann man nun alle weiteren Zeiten bilden, die restlichen Merkmale werden durch Affixe bzw. Zirkumfixe und/oder Hilfsverben ausgedrückt.

STARKE VERBEN Die starken Verben kennzeichnet ein Vokalwechsel im Stammmorphem bei der Bildung von Präteritum und zusätzlich das Zirkumfix *ge-en* beim Partizip II (z.B. *singen – s**a**ng – **ge**sung**en***). Im Deutschen gibt es heutzutage noch etwa 170 starke Stammmorpheme; welche Vokalwechsel bei welchen Verben statt finden, muss jedoch gelernt werden und obliegt keiner offensichtlichen Logik.

SCHWACHE VERBEN Im Gegensatz dazu werden die schwachen Verben ohne Vokalwechsel gebildet, sondern durch das Suffix *te* im Präteritum und das Zirkumfix *ge-et* bzw. *ge-t* im Partizip II (z.b. *kaufen – kauf**te** – **ge**kauf**t***). Es gibt im Deutschen weitaus mehr schwache als starke und unregelmäßige Verben. Von fremden Sprachen entlehnte Verben werden schwach konjugiert, auch finden weitaus mehr Wechsel von der starken zur schwachen Konjugation als umgekehrt statt.

UNREGELMÄSSIGE VERBEN Bei den unregelmäßigen Verben kann sowohl ein Vokalwechsel als auch das Zirkumfix *ge-et* bzw. *ge-t* zur Bildung notwendig sein (z.B. *brennen – br**a**nnte – **ge**brann**t***).

Es werden im folgenden lediglich die starken und schwachen Verben ausführlicher behandelt, da dieses Kapitel auf Grund der vielfältigen Unregelmäßigkeiten zu sehr ausufern würde.

Eine spezielle Ausnahme der unregelmäßigen Verben ist *sein*, da hier der gesamte Stamm wechselt (vgl. *sein – war – geworden*).

INFINITE FORMEN Es existieren drei infinite Formen: der Infinitiv, das Partizip I und das Partizip II; diese sind

allesamt unbestimmt hinsichtlich der Person. Der Infinitiv wird durch den Verb-Stamm und das Suffix *en* bzw. *n* gebildet (letzteres bei Stämmen, die auf *er* bzw. *el* enden, wie z.B. *wander**n***, *kegel**n***). Unter gleichen Voraussetzungen wird für das Partizip I das Suffix *end* bzw. *nd* verwendet (z.B. *sing**end***, *wander**nd***, *kegel**nd***).

Das Partizip II wird gebildet wie bereits beschrieben, allerdings gibt es bei komplexen Verben mit einem selbstständigen Morphem (Präposition oder Adverb) als erstem Bestandteil folgendes zu beachten: Wird das Morphem betont (wie in *über einen Fluss übersetzen*), so tritt das *ge* des Zirkumfix *ge-en*, *ge-n*, *ge-et* bzw. *ge-t* zwischen das betonte Morphem und das Stammmorphem (z.B. *Ich bin über den Fluss über**ge**setzt.*). Anderenfalls (wie z.B. *einen Text übersetzen*) wird das *ge* nicht verwendet (z.B. *Er hat einen Text übersetzt.*).

PERSON UND NUMERUS Die Merkmale Person und Numerus werden durch ein Morphem repräsentiert; lediglich im Präsenz und Präteritum Aktiv flektiert das Verb hinsichtlich der Person und nicht das Hilfsverb (da keines vorhanden). Realisiert wird das jeweilige Morphem nur durch verschiedene Suffixe.

Im Präsenz Indikativ werden sowohl bei starken als auch bei schwachen Verben folgende Suffixe verwendet[6]: *e*, *st* bzw. *est*, *t* bzw. *et*, *en*, *t* bzw. *et* und *en*. Im Präteritum unterscheiden sich die Endungen jedoch, denn für starke Verben werden die Suffixe ∅, *st*, ∅, *en*, *t* und *en*, für schwache Verben ∅, *st*, ∅, *n*, *t* und *n* (dito) verwendet.

Der Konjunktiv wird im Präsenz durch das Suffix *e* ausgedrückt, starke und schwache Verben haben die gleichen Suffixe wie die schwachen Verben im Präteritum Indikativ. Bei dem Konjunktiv im Präteritum der schwachen Verben fällt das Suffix *e* mit dem Präteritum-Suffix *te* zusammen und übrig bleibt *te*; die starken Verben lauten wenn möglich im Stammvokal um und haben ebenso wie die schwachen Verben die gleichen Suffixe wie im Präsenz.

IMPERATIV Die Befehlsform flektiert nur hinsichtlich des Numerus. Bei den schwachen Verben gleicht sie im Singular dem Verbstamm ohne Suffix bzw. mit dem Suffix *e* (vgl. *Kauf ein!* oder *Kauf**e** ein!*), im Plural hat sie statt-

6 Angegeben werden die Endungen wie üblich für 1., 2., 3. Person erst im Singular und darauffolgend im Plural.

dessen das Suffix *et* bzw. *t*. Starke Verben verhalten sich ähnlich, allerdings fällt im Singular das *e* bei jenen Verben weg, die in der 2. und 3. Person Singular Präsenz einen *e/i*-Wechsel kennzeichnet (vgl. *Lies das Buch!*). Zunehmend wird allerdings auch gerade bei diesen Verben der Wechsel im Imperativ nicht immer vollzogen (z.B. *Les(e) das Buch!*).

VERBEN AUF *er* BZW. *el* Ähnlich wie bei den Adjektiven ist auch bei den Verben, deren Stamm auf ein unbetontes *er* oder *el* enden (z.B. *liefern* bzw. *lächeln*) ein Wegfall des *e* möglich, wenn das Suffix mit *e* beginnt (z.B. *ich lächele* oder *ich lächle*).

GETRENNT- UND ZUSAMMENSCHREIBUNG Es gibt Verben, die im Infinitiv zusammen geschrieben, bei bestimmter Verwendung im Satz jedoch getrennt geschrieben werden (z.B. *weggehen*: *Ich gehe weg.* aber *. . . , weil ich weggehe.*). Nach [fdS06] bestehen untrennbare Verben aus der Wurzel des Verbs, der der Stamm eines Substantivs, eines Adjektivs oder ein Partikel vorausgeht. Bei trennbaren Verben jedoch geht der Wurzel des Verbs ein Verbzusatz voraus.[7]

2.2.4 *Artikel*

Folgend wird von der traditionellen Sichtweise ausgegangen, nämlich dass es im Deutschen genau zwei Artikel gibt: den bestimmten *der* und den unbestimmten *ein*. Der Definitartikel flektiert im Singular in Kasus und Genus während er im Plural nur im Kasus flektiert. Der Indefinitartikel flektiert dagegen nur im Singular in Kasus und Genus.

2.2.5 *Pronomina*

Für Pronomina wird auch die Bezeichnung **Begleiter-Stellvertreter** verwendet, da diese Wörter als Begleiter (vgl. *jenes Haus*, *irgendeine Tür*) und/oder als Stellvertreter fungieren können (z.B. *Irgendeiner geht.*, *Jemand steht.* aber **Jemand Mann steht.*).

Sie werden allgemein wie folgt unterschieden:

7 Auf offensichtliche Widersprüche dieser Regel wird nicht eingegangen: es heißt z.B. *Der Arzt schreibt sie krank.* und nicht **Der Arzt krankschreibt sie.*, obwohl *krank* ein Adjektiv ist. Ferner sind Verben mit einem Verbzusatz daran erkennbar, dass „die Reihenfolge ihrer Bestandteile in Abhängigkeit von ihrer Stellung im Satz wechselt.“.

PERSONALPRONOMINA Das Pronomen *ich* flektiert hinsichtlich Kasus, Person und Numerus, in der 3. Person Singular sogar im Genus.

REFLEXIVPRONOMINA Das einzige Reflexivpronomen ist *sich*. Es ist unveränderlich und tritt im Akkusativ und Dativ sowohl im Singular als auch im Plural in allen drei Genera auf.

INDEFINITPRONOMINA Diese (autonomen) indefiniten Pronomina haben ein festes Genus, flektieren im Kasus nur im Singular (ausgenommen *man*). Sie lassen sich nach Genus wie folgt einordnen:

MASKULIN jemand, irgendwer, irgendjemand, niemand, jedermann, man

NEUTRUM etwas, irgendwas, irgendetwas, nichts

WER/WAS Meist erfüllt *wer* bzw. *was* die Funktion des Interrogativpronomens. Sowohl *wer* als auch *was* flektieren nur im Singular hinsichtlich Kasus und haben jeweils ein festes Genus (Maskulinum bzw. Neutrum).

DEMONSTRATIVA Als Demonstrativa werden im Allgemeinen *dieser*, *jener*, *derjenige* und *der* (mit betontem *e*) angenommen. Sie flektieren im Singular nach Kasus und Genus, im Plural nur nach Kasus.

Das Demonstrativum flektiert im Plural unterschiedlich, je nachdem ob es adnominal (vgl. *Die Prüfung **der** Studenten.*, gemeint im Sinne von *Die Prüfung dieser Studenten dort.*) oder pronominal (vgl. *Die Prüfung derer.*) verwendet wird.

INDEFINITA Nicht nur pronominal verwendet und daher nicht zu den Indefinitpronomina sondern zu den Indefinita gehören *aller*, *einiger*, *etlicher*, *jeder*, *jedweder*, *jeglicher*, *irgendein*, *irgendwelcher*, *kein*, *mancher* und *mehrere*.

Es flektieren wie *dieser*: *aller*, *einiger*, *etlicher*, *irgendwelcher*, *jedweder*, *jeglicher* und *mancher*. Nur im Singular existiert analog *jeder*, nur im Plural *mehrere*.

Je nach adnominaler oder pronominaler Verwendung flektieren *kein* und *irgendein* jeweils unterschiedlich.

POSSESIVA Die Possesiva *mein*, *dein*, *sein*, *ihr*, *unser*, *euer* und *ihr* flektieren wie *kein*, auch unterschiedlich je nach Verwendung.

WELCHER *Welcher* fungiert hauptsächlich als Interrogativum; es kann adnominal (vgl. *Welchen Mantel soll ich anziehen?*), aber auch pronominal (vgl. *Welchen soll ich anziehen?*) verwendet werden. Daneben kommt es auch mit indefiniter Bedeutung ausschließlich pronominal vor (vgl. *Hat er noch Blätter? – Ja, er hat welche.* aber **Ja, er hat welche Blätter.*).

Es wird flektiert wie *dieser*, tritt aber auch unflektiert auf (wie *mancher*): z.b. in *Welch schöne Aussicht!*.

RELATIVPRONOMINA Als Relativpronomina werden verwendet *der*, *welcher* und *wer/was*. Sie flektieren allesamt wie bei ihrem jeweiligen anderen Gebrauch, also *der* wie das entsprechende Demonstrativpronomen, *welcher* und *wer/was* wie das jeweils entsprechende Interrogativum.

2.2.6 *Adverb*

Adverbien bestimmen ein Verb, ein Partizip, einen Satz oder ein Adjektiv näher. Wenige Adverbien können kompariert werden (z.B. *oft*, *gern*).

Die Suffixe für die Steigerung sind analog zu denen der Adjektive, auch tritt Umlautung auf; der Großteil der Adverbien hat in Komparativ und Superlativ einen anderen Wortstamm als im Positiv (vgl. [KRSSW07]).

2.2.7 *Fremdwörter*

Es gibt einige vor allem aus dem Griechischen und Lateinischen übernommene Wörter, die nicht flektieren wie bisher beschrieben (z.B. *die Bronchitis, die Bronchitiden*); es finden sich allerdings auch Wörter, deren Flexion regelmäßig sein kann (z.b. *das Taxi: die Taxis* oder *die Taxen*). Erwähnenswert ist, dass der Nominativ Singular einiger Substantive nicht unmarkiert ist, sondern durch ein Suffix markiert wird (z.B. *das Cell**o**, die Celli*).

2.3 FLEXION IN ANDEREN SPRACHEN

Eine Ziel bei der Implementierung von FLEXI ist die Sprachunabhängigkeit. Daher soll nach dem ausführlichen Überblick über die Flexion des Deutschen auch kurz auf die Flexion in anderen Sprachen eingegangen werden, die nach einem anderen Grundsatz funktioniert als der Affigierung und Stammalternierung.

2.3.1 *Wurzel- und Mustermorphologie*

In den semitischen Sprachen findet bei der Flexion oft eine Veränderung von vokalischen Mustern statt, während die konsonantische Wurzel erhalten bleibt.

Im Arabischen wird z.B. der Singular bzw. der Plural der Wurzel *bnk* (zu übersetzen mit *Bank*) als *bank* bzw. *bunuuk* realisiert. Die konsonantische Wurzel bleibt also erhalten, während ein Muster von Vokalen den Numerus ausdrückt.

2.3.2 *Reduplikation*

Eine weitere Art zu flektieren ist in bestimmten Sprachen die Reduplikation, bei der der Anfang oder das Ende eines Wortes oder auch das ganze Wort wiederholt wird. Im Indonesischen z.B. wird der Plural von *kuda* (zu übersetzen mit *Pferd*) als *kuda-kuda* realisiert. Im Gegensatz dazu wird im Ilokano der Plural von *trák* (zu übersetzen mit *Lkw*) als *tra:-trák* realisiert. Die Reduplikation hängt stark mit der Wurzel- und Mustermorphologie zusammen, da die wiederholten Teile bestimmten Mustern folgen können.

Für weitere Informationen vgl. [MA05, S. 76-78, 166-168] und [Mar82].

2.4 SCHLUSSFOLGERUNGEN

Nach den vorangegangenen Beobachtungen ist folgendes festzuhalten:

NICHT-KOMPARIERBARKEIT Es wurde erläutert, durch welche Kriterien ein Adjektiv als nicht-komparierbar eingestuft wird. Diese Eigenschaft verliert das Adjektiv jedoch, wenn es nicht in seinem ursprünglichen Sinn gebraucht wird. So kann *tot* gesteigert werden, wenn es in dem Kon-

text nicht um den Tod eines Lebewesens geht (z.B. um die Eigenschaft *tot* des Nachtlebens einer Stadt). Also werden in der Implementierung dieser Arbeit alle eigentlich nicht-komparierbaren Adjektive gesteigert und es muss kontext-abhängig (vom Benutzer) entschieden werden, ob eine Steigerung sinnvoll ist oder nicht.

NORMEN BEI DER BILDUNG Da Anwendungsfälle wie automatische Rechtschreibkorrektur und Wortvervollständigung das Ziel sind, sollte dieses System auf gewisse Normen nicht verzichten. Da das Wissen des Systems einzig aus der Trainingsmenge stammt, ist es also notwendig auf die Korrektheit der Wörter der Trainingsmenge zu achten.

WORTVERÄNDERUNG Wie schon erwähnt, werden bei der Implementierung dieser Arbeit lediglich zwei Ebenen betrachtet: Einerseits die Grundform und andererseits die flektierte Form; daher werden nicht verschiedene Zwischenstufen (wie z.B. Phonologie, Orthographie, etc.) betrachtet, die auf die Wortform Einfluss haben (siehe hierzu Abschnitt 4.2 auf Seite 43).

Zudem sollen aber nur die Teile eines Wortes betrachtet werden, die sich verändern; so wird nun die Veränderung von *Schloss* zu *Schlösser* aufgefasst als:

1. Der Stammvokal wird ersetzt mit *ö*.
2. An das Wortende wird *er* angefügt.

Würde man diese Veränderung nicht so abstrakt formulieren, sondern konkreter als eine atomare Operation wie z.B. *Die letzten 3 Zeichen löschen und „ösֲser“ anfügen*, so könnte man dies nicht auf z.B. *Holz* anwenden, obwohl *Holz* und *Schloss* gleich flektieren; dies ist in genau dem Fall wichtig, bei dem klar ist, dass Holz und Schloss gleich flektieren, aber FLEXI nur das eine Wort, aber nicht das andere bekannt ist.

Allgemein gesehen finden also folgende Veränderungen statt:

SUFFIGIERUNG Wie z.B. das *s* bei *Auto**s***.

PRÄFIGIERUNG Wie z.B. das *ge* bei ***ge**sehen*.

LÖSCHEN Wie z.B. das *c* bei *höher*.

ERSETZEN Wie z.B. das *ö* bei *Kl**ö**ster*.

EINFÜGEN Wie z.B. das *ge* bei *ab**ge**fahren*.

VERSCHIEBEN Wie z.B. das *weg* bei *fahre* ***weg*** bzw. das *fahre* bei ***fahre*** *weg*.

MERKMALE Zusammengefasst kann die Zuordnung zu einer Flexionskategorie beeinflusst werden durch: die Ursprungssprache, die Wortart, das Geschlecht, die Betonung einer bestimmten Silbe, Endung der Grundform, Zugehörigkeit zu einer Bedeutungskategorie (wie z.B. Lebewesen), die Anzahl der Silben, der Präfix, Fähigkeit des Stammvokals umzulauten, der Wortstamm[8] und die Grundform selbst.[9] Zudem gibt es Elemente wie das *e* in *des Mann**e**s*, deren Verwendung fakultativ ist.

8 Wird eine Kategorie nur durch verschiedene Wortstämme definiert, spricht im Allgemeinen von Ausnahmen.

9 Natürlich wird nicht die Flexion aller Grundformen durch alle Merkmale bestimmt.

3

SOFTWARE ZUR FLEXION

Nachdem ausführlich die Flexion des Deutschen beschrieben wurde, folgt nun eine Übersicht über die frei erhältlichen Programme, die in der Lage sind, Grundformen des Deutschen zu flektieren. Diese Übersicht ist keine ausführliche Evaluierung der Leistung und Qualität der Programme, sondern setzt sich vielmehr mit den Datenstrukturen zur Speicherung der Daten zur Flexion auseinander und beschreibt den Ablauf der Generierung der flektierten Wortformen. Zudem folgt jeweils ein Fazit basierend auf einem Kurztest.

3.1 MORPHY

Morphy ist unter der Leitung von Dr. Wolfang Lezius entstanden und ein umfangreiches Softwarepaket, im Folgenden wird lediglich der Teil von Morphy beschrieben, der flektierte Wortformen generiert (vgl. [Lez96]).

3.1.1 *Spezifikation*

Jedem Stamm wird eine Klasse zugeordnet; die Klassen werden repräsentiert durch je eine Zeile in einer wortartspezifischen Tabelle, in der Flexionsendungen, das Genus und eventuell die Umlautung festgelegt sind. Ferner wird noch vermerkt, ob ein ß-ss-Wechsel vorliegt (z.B. *Kuß - Kusses*) oder ob ein Wort nicht im Plural vorkommt (z.B. *Laub*).

Die Flexionsendungen können optionale Elemente enthalten (z.B. *s/es* im Genitiv Singular von *Wolf*), aber auch das leere Element -, also keine Endung.

Für nicht-schwache Verben werden lediglich sieben Formen gespeichert, aus denen sich alle restlichen Formen ableiten lassen.

3.1.2 *Ableitung der flektierten Formen*

Die Generierung der flektierten Formen erfolgt in drei Schritten:

1. Die eingegebene Grundform wird auf Flektierbarkeit untersucht. Die nicht-flektierenden Formen sind in einem Lexikon gespeichert; zu den flektierenden Formen zählen für Morphy Substantive, Adjektive, Verben und Eigennamen.
2. Ist eine Form flektierbar, wird der Stamm identifiziert.
3. Ist diesem Stamm im Lexikon ein Eintrag zugeordnet, können die Flexionsdaten generiert werden.

3.1.3 *Fazit*

Morphy weist folgende Nachteile auf:

- Die Regeln zur Generierung der flektierten Wortformen sind starr im Quellcode verankert, der nicht einsehbar ist.
- Eine Portierbarkeit auf andere Sprachen ist nicht vorgesehen.
- Man kann zwar Worte zum Lexikon hinzufügen und wird dabei von dem System unterstützt, allerdings ist dies nur manuell möglich.
- Ein Schnelltest ergab, dass das Lexikon einige Lücken aufweist.[1]
- Man kann das System lediglich über die grafische Benutzeroberfläche ansprechen; eine Integrierung in eine eigene Software ist also nur schwer möglich.

Dennoch seien auch folgende Vorteile erwähnt: Die grafische Benutzeroberfläche ist recht Benutzerfreundlich, auch lassen sich gut größere Mengen von Wörtern flektieren, da die Möglichkeit besteht, Morphy eine Datei zu übergeben,

1 Getestet wurden die Wörter *Bank*, *Saal*, *oft*, *hoch*, *viel*, *gut*, *groß*, *super*, *pappsatt*, *satt*, *wer*, *dieser*, *gesund*, *Landesinnenminister*, *Minister*, *weggehen* und *Tisch*. Vollständig und korrekt gebeugt wurden lediglich *Saal*, *pappsatt*, *satt*, *Minister* und *Tisch*.

die eine Liste von Wörtern enthält. Die Ausgabe geschieht wiederum in eine Datei, so dass die Daten maschinell weiterverabeitet werden können.

3.2 LA-MORPH

In der Dokumentation (vgl. [Hau]) wird LA-Morph als Teilprojekt von JSLIM erläutert, desweiteren wird eine Beispielimplementierung für ein Lexikon mit wenigen Worten beschrieben.

3.2.1 *Spezifikation*

Zunächst wird ein Grundformenlexikon definiert, das Angaben zu einer bestimmten Grundform enthält, z.B. die zugehörige Wortart. Weiterhin werden Repräsentationen für die jeweiligen Worte eingetragen, gegeben ist das Beispiel die Repräsentation *TUch* für *Tuch*, d.h. wenn der Benutzer *TUch* eingibt, werden ihm automatisch die flektierten Formen von *Tuch* angezeigt. Zur Generierung der flektierten Formen wird eine Datei benötigt, in der für die Grundformen durch reguläre Ausdrücke die verschiedenen benötigten Stämme beschrieben werden (nach dieser Theorie hat z.B. *essen* die Stämme *ess*, *iss* und *äß*, in dieser Datei steht demnach folgendes: */(ess)en/* und die Stämme */$1/*, */iss/* und */äß/*). Die dazugehörigen Flexionsendungen stehen in weiteren (nach Wortarten Nomen, Adjektiv und Verb) getrennten Dateien.

3.2.2 *Fazit*

LA-Morph enthält Daten für ca. 96000 Verben, 117000 Substantive und 12000 Adjektive. Ist ein Wort nicht im Lexikon eingetragen, können keine Daten hierfür geliefert werden, eine Flexion von unbekannten Wörtern ist somit nicht möglich. Auf Nachfrage wurde das Programm samt der Quellen und Ressourcen zur Verfügung gestellt, ist aber nicht online verfügbar. Die Online-Demo[2] funktionierte im Juli 2009 nicht.

2 Verfügbar online unter `http://www.linguistik.uni-erlangen.de/clue/de/forschung-projekte/jslim/online-demo-2.html`

3.3 MORPHIX

Morphix wurde bereits 1986 in Lisp implementiert (vgl. [FN88]); es gibt eine Reimplementierung (Morphix++) von 1995 (vgl. [Neu]) und ist samt Ressource-Dateien unter einer eigenen Lizenz verfügbar.

3.3.1 *Spezifikation*

Leider ist die Dokumentation nicht konkret genug; auch die Datenstrukturen zur Speicherung sind weder dokumentiert noch kommentiert. In der recht unvollständigen Dokumentation findet sich einzige folgende Aussage: „Thus, the basic processing strategy employed by MORPHIX++ consists of trie traversal combined with the application of finite state automata“.[3]

3.3.2 *Fazit*

Auch Morphix ist nicht lernfähig; die Bedienungsweise kann nicht gerade als benutzerfreundlich bezeichnet werden: Die Bedienung findet über den Lisp-Interpreter statt, zudem müssen bei der Generierung die passenden morphosyntaktischen Merkmale mit angegeben werden. So sieht z.B. der Aufruf zur Generierung von der Plural-Form von *Haus* wie folgt aus:

```
1 (generate "haus" 'noun :number 'pl)
```

Ausgabe ist „Haeuser“.

Weiterhin ist das Lexikon mit ca. 120000 Stämmen erwähnenswert; betont wird in der Dokumentation außerdem die hohe Geschwindigkeit (5000 Wörter pro Sekunde ohne, 2800 Wörter pro Sekunde mit Behandlung von Komposita).

3.4 VERBFORMEN

Eine webbasierte Software, die lediglich Verben (des Deutschen) konjugiert, ist VERBformen.[4] Es gibt hierzu keine

3 Übersetzt ins Deutsche etwa: „Folglich besteht die grundlegende Strategie der Abarbeitung von MORPHIX++ aus Traversierung von Bäumen kombiniert mit der Anwendung eines endlichen Zustandsautomaten“.

4 Zu erreichen unter `http://www.verbformen.de`.

wissenschaftlichen Veröffentlichungen, die Informationen auf der Webseite sind relativ dürftig.

3.4.1 *Funktionsweise*

VERBformen nutzt keine digitalen Wörterbücher, sondern generiert für einen Infinitiv alle Formen des Deutschen nach Bildungsregeln. Es können also auch Verbneuschöpfungen wie z.B. *googlen*, Entlehnungen aus anderen Sprachen wie z.B. *updaten* oder unpersönliche Verben wie z.B. *hageln* eingegeben werden.

3.4.2 *Kurztest*

VERBformen wurde einem Kurztest unterzogen, der zeigen soll, ob die Angaben auf der zugehörigen Webseite stimmen:

- Die unpersönlichen Verben *hageln*, *schneien*, *regnen* und *blitzen* wurden korrekt gebeugt (jeweils nur z.B. *es schneit* und nicht **ich schneie*).
- Für das mehrdeutige Verb *umfahren* werden zunächst nur die Formen des Bedeutungsaspekts *fahrend anstoßen* und *zu Boden werfen* generiert. Man gelangt allerdings durch einen Verweis zu den Formen des Bedeutungsaspekts *um etwas herumfahren*.
- Das Verb *sterben* wird über generiert, da auch Formen wie z.B. *?ich war gestorben worden* oder *?ich war gestorben gewesen* erzeugt werden, die zumindest fragwürdig sind.
- Für das aus dem Englischen übernommene Verb *downloaden* wird der Perfekt falsch gebildet: z.B. **ich habe downloadet* statt *ich habe downgeloadet* (vgl. [KRSSW07]).
- Die Verbneuschöpfung *googeln* wird korrekt gebeugt, ebenfalls das unregelmäßige Verb *sein*.

3.4.3 *Fazit*

Als die einzige nicht-wissenschaftliche der beschriebenen Anwendungen ist VERBformen von recht guter Qualität

und für die Anwendung geeignet. Es ist zwar nur auf Verben begrenzt und es sind einige Fehler zu finden, jedoch stechen die Benutzerfreundlichkeit, der recht gute Umgang mit unbekannten und auch aus dem Englischen übernommenen Verben positiv hervor. Es gibt allerdings sehr wenig Dokumentation, so dass sich nicht wie bei den anderen Anwendungen weitere Analysen anstellen lassen.

3.5 CANOONET

CanooNet ist eine weitere webbasierte Software.[5] Sie wird betrieben und finanziert von der Canoo Engineering AG in Basel. Entstanden ist die Software in langjähriger Zusammenarbeit zwischen Mitarbeitern der Universität Basel, der Vrije Universiteit Amsterdam, des IDSIA Lugano und der Canoo Engineering AG. Hauptverantwortlicher im wissenschaftlichen Bereich ist laut Webseite Dr. Stephan Bopp, derzeit Linguist an der Universität Zürich.

Die Webseite wird durch Verwendung von WMTrans-Produkten ermöglicht, diese wiederum basieren auf den WordManager-Datenbanken. Die Quellen der Software sind nicht frei zugänglich, ebenso gibt es keine Publikationen darüber, wie die Flexionstabellen funktionieren.

3.5.1 *WMTrans Inflection Generator*

Der *Inflection Generator* existiert in zwei unterschiedlichen Versionen: einmal als Implementierung in Java, andererseits als plattformspezifische Software mit je einer API in ANSI C/C++ und Java. Erhältlich sind Daten für das Englische, Deutsche und Italienische; für das Deutsche enthalten die Daten über 210.000 Wörter.[6]

Als Beispiel für die Ein- bzw. Ausgabe ist auf der Webseite folgendes zu finden:

```
1 query   -> haus
2 result  -> [...]
3    haus
4      (Cat N)(Gender N)(Num SG)(Case Nom)(ID 0-1),
```

5 Zu erreichen unter `http://www.canoo.net`.

6 Die Preise erfährt man auf Nachfrage: die Lizenz pro Jahr und CPU für den *Inflection Generator* kostet inklusive der Daten für das Deutsche 3000,- Euro. Die Daten zur Generierung sind jedoch nicht im Klartext beigefügt, sondern in verschlüsselter Form. Will man die Liste der deutschen Grundformen inklusive morphologischen Zusatzinformationen im Klartext beziehen, kostet dies 50.000,- Euro.

```
5       (Cat N)(Gender N)(Num SG)(Case Dat)(ID 0-1),
6       (Cat N)(Gender N)(Num SG)(Case Acc)(ID 0-1),
7     hause
8       (Cat N)(Gender N)(Num SG)(Case Dat)(ID 0-1),
9             [...]
```

Zu sehen sind die unterschiedlichen Wortformen mit ihren jeweiligen Merkmalen.

Die Datenhaltung der CanooNet Anwendungen geschieht mittels der WordManager-Datenbanken.

3.5.2 *WordManager*

WordManager ist eine Software für die Erstellung von Wörterbüchern mit morphologischen Informationen. Zunächst fängt ein Linguist damit an, Informationen in eine leere Datenbank einzugeben, die dafür vorbereitet wurde, verschiedene Typen von Informationen verarbeiten zu können. Die Aufgabe des Linguisten besteht konkret darin, Regeln für die Flexion und die Wortbildung anzugeben, inklusive die Beispiele für die Anwendung und die Beschreibung von irregulären Prozessen. Die darauf folgende Aufgabe des Lexikographen ist es, nun ein Wörterbuch mit Einträgen zu erstellen, die sich auf die bereits vorhandenen Regeln beziehen. So kann von dem Lexikographen z.B. für *self-determination* (engl. für *Selbstbestimmung*) spezifiziert werden, dass dieses Wort aus zwei Teilen besteht. Falls *determination* noch nicht im Wörterbuch vorhanden ist, müssen die weiteren Wortbestandteile analysiert werden. Für weitere Informationen vgl. [DtH92].

3.5.3 *Fazit*

Bei CanooNet handelt es sich um die einzige der bisher beschriebenen Anwendungen, die aktuell gepflegt wird und auf dem Stand der Neuen Deutschen Rechtschreibung ist. Zudem ist es die (subjektiv gesehen) benutzerfreundliche und ansprechendste Software mit einem recht großen Umfang an weiteren Funktionen (z.B. Informationen zur Wortbildung, ein Glossar mit Fachbegriffen, Angaben zur Rechtschreibung etc).

Erwähnenswert ist weiterhin, dass die Daten allesamt manuell aus verschiedenen Quellen zusammengetragen wurden und jeder Datensatz einzeln durch ein Lexikographenteam den Weg in die Datenbanken gefunden hat. So-

mit ist zwar ein gewisser Anspruch an Qualität gewährleistet, ob diese jedoch im Verhältnis zu den Kosten steht, bleibt fragwürdig.[7]

Für unbekannte Wörter, wie z.B. *Kapitänsmütze*, findet keine automatisierte Generierung der flektierten Formen statt; allerdings erkennt die Software *Unknown Word Analyzer* die Bestandteile *Kapitän*, Fugen-Element *s* und *Mütze*, durch die man per Link zu den Wortformen von *Mütze* gelangen kann.

3.6 GESAMT-FAZIT

Es gibt zahlreiche Programme, die in den Bereich der Morphologie-Werkzeuge gehören, jedoch nur eine Hand voll solcher, die flektierte Wortformen generieren können. Diese können teilweise auf eine lange Entwicklungszeit zurückblicken, das älteste (Morphix) stammt immerhin von 1984, auch 15 Jahre nach der Reimplementierung ist Morphix immer noch auf einem aktuellen PC lauffähig.

Dennoch bedingt das Alter einige grundlegenden Schwächen: Sämtliche Datenstrukturen sind darauf ausgelegt, Speicherplatz zu sparen und zudem teilweise noch auf die Geschwindigkeit zum Auffinden einer gegebenen Grundform hin optimiert. Bei den heutigen Preisen für Speichermedien und der aktuellen Rechenleistung von Computern dürfte dies bei einer Neuentwicklung kaum eine Rolle mehr spielen.

Bei der einzig aktuellen Entwicklung (CanooNet) wird die Schonung der Ressourcen Speicher und Zeit nicht hervorgehoben, sondern vielmehr die Qualität und der Umfang der Daten, aus denen sich flektierte Wortformen generieren lassen.

Ebenso ist CanooNet eine Anwendung, bei der sich eine unbekannte Grundform über bekannte Grundformen erschließen lässt. Gibt der Nutzer z.B. *Schweinebratenkruste* ein, wird dieses Wort in *Schweinebraten* und *Kruste* zerlegt, per Mausklick gelangt der Nutzer dann zu den Wortfor-

7 Bisher hat keine Evaluation statt gefunden, zumindest ist bislang keine Publikation dazu erschienen. Es ist auch nicht möglich, dies einfach zu tun, Zitat aus den Nutzungsbedingungen (erreichbar online unter `http://www.canoo.net/services/ueberblick/nutzung.html`):

> Es ist insbesondere nicht gestattet, jedwede Art von Inhalten per Script oder auf andere Weise automatisiert abzufragen.

men von *Kruste*. VERBformen ist eine weitere Anwendung, mit der ebenfalls ihr unbekannte und neue Wörter flektiert werden können. Bei dieser Anwendung wird nach Regeln und nicht auf Basis eines Lexikons konjugiert.

4

ANGEWANDTE VERFAHREN

In dieser Arbeit soll ein Verfahren entwickelt werden, mit dem bei Eingabe einer Grundform deren flektierte Formen ausgegeben werden können. Hierfür wird auf Basis einer Menge von bereits flektierten Wörtern festgestellt, welche Veränderungen durch welche Merkmale bestimmt werden. Abstrakt beschrieben sind also folgende Schritte zu gehen:

1. Erstellen der Trainingsmenge:
   a) Liste der flektierten Formen, der jeweiligen Grundform und der zugehörigen morphosyntaktischen Merkmale erstellen.
   b) Die Merkmale der Grundformen ermitteln, die für die Flexion bestimmend sind.
2. Erstellen der flektierten Formen einer Grundform:
   a) Aus der Trainingsmenge lernen, welche Merkmale zu welcher Art der Flexion führen.
   b) Ermitteln der Merkmale der eingegebenen Grundform.
   c) Mit diesem Wissen die Grundform flektieren und die Formen (unter Angabe der morphosyntaktischen Merkmale) zurückgeben.

Im Folgenden werden nun die hierfür notwendigen Verfahren beschrieben.

4.1 ERSTELLEN DER TRAININGSMENGE

Vor dem Beginn dieser Arbeit wurden bereits eine Menge von flektierten Wortformen erstellt; hierbei handelt es sich um ca. 835.000 unterschiedliche Grundformen und 2.500.000 unterschiedliche flektierte Formen, wobei *Haus* (Nominativ Singular) und *Haus* (Dativ Singular) als eine Form gezählt wurden. Bei dem Erstellen der Daten wurde weitgehend auf Vollständigkeit geachtet, das heißt,

dass für jede mögliche Kombination morphosyntaktischer Merkmale auch mindestens eine Wortform existiert. Ferner wurde darauf Wert gelegt, dass die flektierte Form jedes Datensatzes (bestehend aus Grundform, flektierter Form und Bezeichnung für die Merkmalskombination) eindeutig ist[1]. Als Bezeichnungen für die morphosyntaktischen Merkmale wurden Abkürzungen verwendet, z.B. *NomSg* (Nominativ Singular) oder *NomSgNeutbArt* (Nominativ Singular Neutrum bestimmter Artikel). Prinzipiell ist die Wahl dieser Bezeichner nur so zu treffen, dass eindeutig ist, welcher Bezeichner welche Merkmalskombinationen meint.

4.1.1 *Ermitteln der Merkmale*

Wie der letzte Buchstabe einer Grundform und die Grundform als Merkmal selbst zu ermitteln sind, wird nicht näher erläutert. Der Algorithmus hierzu ist offensichtlich.

Compact Patricia Trie

Zunächst folgt eine Erläuterung des *Compact Pratricia Trie* (CPT), da dieser im Folgenden mehrfach genutzt wird, z.B. für das Bestimmen der Wortart einer Grundform.

Ein Trie ist spezieller Graph. Ein **Graph** G wird nach [Die06] definiert wie folgt:[2]

$$G = (E, V), V \cap E = \emptyset, E \subseteq V \times V \tag{4.1}$$

E ist die Knotenmenge, V die Kantenmenge von G. G heißt **endlich** bzw. **unendlich**, falls V endlich bzw. unendlich ist.

Sei $G = (E, V)$ Graph. G heißt **gerichtet**, falls gilt

$$\forall u, v \in V : (u, v) \in E \rightarrow (v, u) \in E, \tag{4.2}$$

anderenfalls **ungerichtet**.

Sei $G = (E, V)$ gerichtet und $k = (v_0, \ldots, v_n) \in V^{n+1}$.

1 Dies ist deswegen zu erwähnen, da es Software gibt, die Einträge der Art *anpummele,anpummle/pummele,pummle an* erzeugt. Hieraus sind vier Formen zu extrahieren: *anpummele*, *anpummle*, *pummele an* und *pummle an*.

2 Hier wird bewusst auf die Definition von Kanten als zwei-elementige Teilmengen aus V verzichtet, um später problemlos gerichtete und ungerichtete Graphen unterscheiden zu können.

k heißt **Kantenfolge** der Länge n von v_0 nach v_n, falls gilt

$$\forall i \in \{0, \ldots, n-1\} : (v_i, v_{i+1}) \in E \tag{4.3}$$

$v_1, \ldots, v_{n-1}$ sind die **inneren Knoten** von k. Ist $v_0 = v_n$, so ist k **geschlossen**.

k heißt **Kantenzug** der Länge n von v_0 nach v_n, wenn k Kantenfolge der Länge n von v_0 nach v_n ist und gilt

$$\forall i, j \in \{0, \ldots, n-1\} : i \neq j \rightarrow (v_i, v_{i+1}) \neq (v_j, v_{j+1}) \tag{4.4}$$

.

k heißt **Weg** der Länge n von v_0 nach v_n, falls k Kantenfolge der Länge n von v_0 nach v_n ist und wenn gilt

$$\forall i, j \in \{0, \ldots, n-1\} : i \neq j \rightarrow v_i \neq v_j \tag{4.5}$$

k heißt **Zyklus** der Länge n, falls k geschlossene Kantenfolge der Länge n von v_0 nach v_n ist und wenn

$$k' = (v_0, \ldots, v_{n-1}) \tag{4.6}$$

ein Weg ist. Existiert in G kein Zyklus, so heißt er **kreisfrei**.

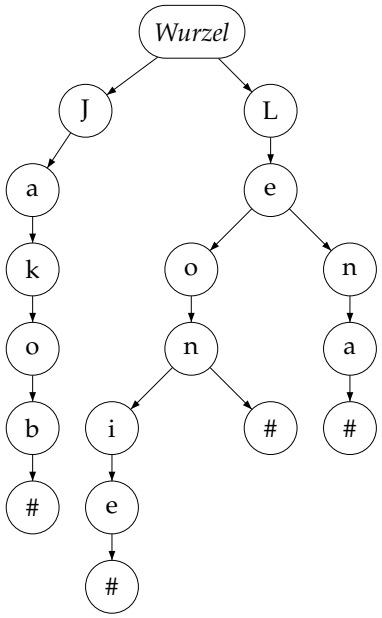

ABBILDUNG 4.1: Beispiel für einen Trie, in dem Vornamen gespeichert sind.

Sei $G = (E, V)$ ungerichteter bzw. gerichteter Graph. G heißt **zusammenhängend**, falls zwischen je zwei Knoten ein Kantenzug existiert bzw. falls in dem dazugehörigen ungerichteten Graphen[3] zwischen je zwei Knoten ein Kantenzug existiert.

Sei $G = (E, V)$ ein Graph. Der **Eingangsgrad** bzw. **Ausgangsgrad** eines Knotens ist als

$$eg(v) = |\{u \mid (u, v) \in E\}| \tag{4.8}$$

bzw.

$$ag(v) = |\{u \mid (v, u) \in E\}| \tag{4.9}$$

3 Also der Graph $G' = (E', V)$ mit

$$E' = E \cup \{(v, u) \mid (u, v) \in E\} \tag{4.7}$$

.

definiert.

Sei $G = (E, V)$ ein gerichteter Graph. G heißt **gerichteter Wald**, falls G keinen Zyklus besitzt und gilt

$$\forall v \in V : eg(v) \leqslant 1 \tag{4.10}$$

Die Knoten $u : eg(u) = 0$ des Waldes sind **Wurzeln**.

Ein **gerichteter Baum** (weiterhin schlicht **Baum**) ist ein gerichteter Wald mit genau einer Wurzel.

Die Knoten

$$u : eg(u) = 1, ag(u) = 0 \tag{4.11}$$

in einem Baum heißen Blätter.

Sei $G = (E, V)$ Baum und $(u, v) \in E$. u heißt dann Vorgänger von v, v Nachfolger von u.

In [HQW06, S. 71 ff] wird der **Trie** (nach *Information Re**trie**val*) als Datenstruktur beschrieben, die es ermöglicht effizient Wörter zu speichern und darin nach ihnen zu suchen. Tries sind eine Unterart der **m-Wege-Bäume**, also Bäume für die gilt

$$\forall u \in V : ag(u) \leqslant m \tag{4.12}$$

Das m wird bestimmt durch die Anzahl der Buchstaben des Alphabets aus dem die zu speichernden Wörter stammen. Im Wurzelknoten wird gespeichert, mit welchem Buchstaben die Wörter im Trie beginnen, auf Ebene i findet sich dann der i. Buchstabe der Wörter (siehe Abbildung 4.1 auf S. 37).

Ein Patricia Trie (nach [Mor68]) ist nun ein spezieller Trie: einerseits können zusätzliche Informationen an den Knoten gespeichert werden, andererseits kann ein Knoten mehrere Buchstaben enthalten. Haben für ein Merkmal x i Blätter, die von einem Knoten u zu erreichen sind, das Merkmal a, so kann dies direkt am Knoten u inklusive der Anzahl gespeichert werden.

Aus dem Patricia Trie wird nun ein Compact Patricia Trie (CPT), indem überflüssiges gelöscht wird: redundante Verzweigungen und Blätter, die mehr als 1 Zeichen enthalten, können gestutzt werden (von dem Englischen *pruning*). So verliert der Trie zwar an Größe, die Aussagen, die daraus abgeleitet werden können, sind jedoch identisch mit den Aussagen aus dem Baum, der nicht gestutzt wurde. Ein Beispiel für einen Patricia Trie ist mit Abbildung 4.2

auf Seite 39 gegeben, ein Beispiel für einen Compact Patricia Trie mit Abbildung 4.3 auf Seite 41.

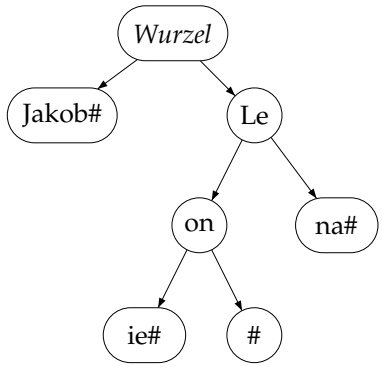

ABBILDUNG 4.2: Beispiel für einen Patricia Trie ohne Klassifikation zur Speicherung von Vornamen

Will man nun herausfinden, welcher Wert eines Merkmals mit welcher Wahrscheinlichkeit auf ein bestimmtes Wort zutrifft, schreitet man die Knoten des Baumes solange ab, wie es keinen Mismatch gibt, und verwendet zur Berechnung die Merkmale in dem so gefundenen Knoten.

Schnell ersichtlich ist, dass diese Art der Berechnung nicht für beliebige Merkmale funktionieren kann, sondern nur für solche, die sich (mehrheitlich) auf Wortenden oder -anfänge zurückführen lassen ([HQW06, S. 76] nennen z.B. grammatikalisches Geschlecht, Numerus und Wortart). In dem CPT in Abbildung 4.3 auf Seite 41 ist das natürliche Geschlecht der Vornamen gespeichert.

Will man für den Vornamen *Justus* das Geschlecht herausfinden, endet die Suche bei dem Knoten mit dem Inhalt *J* und es wird der Wert *m* mit der Wahrscheinlichkeit 1 ermittelt. Klar ist, dass nicht jedes Ergebnis so eindeutig sein muss, da z.B. *Kim* sowohl männlich als auch weiblich sein kann.

Für den CPT wird durchweg die Implementierung der Abteilung der *Automatischen Sprachvearbeitung* (ASV) der Universität Leipzig verwendet.[4]

Der CPT wird für die Ermittelung zweier Merkmale verwendet: für das Geschlecht von Substantiven und die Wortart von Grundformen. Dies wird in den folgenden zwei Abschnitten kurz näher beschrieben.

GESCHLECHT VON SUBSTANTIVEN Für die Bestimmung des Geschlechts eines Substantivs wird ein CPT verwendet, dessen Trainingsmenge ca. 835000 Grundformen und deren (grammatisches) Geschlecht enthält. Selbstverständlich werden die Wörter rückwärts in den CPT eingefügt, so dass der Suffix und nicht der Präfix ausschlagge-

4 Zu finden unter `http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/Pretree.html`.

bend ist. Ein Auszug aus den Trainingsdaten findet sich in Abschnitt A.1 auf Seite 83.

WORTART VON GRUNDFORMEN Auch hierzu wurde ein CPT verwendet, dessen Trainingsmenge aus ca. 833000 Grundformen und deren Wortart enthält, ebenso werden die Wörter rückwärts eingefügt. Ein Auszug aus den Trainingsdaten findet sich in Abschnitt A.2 auf Seite 84.

Die Wurzel einer Grundform

Wörter flektieren wie ihre Wurzel[5]. [6] Deshalb ist die Wurzel ein wichtiges Merkmal. [BW05, S. 4] beschreiben ein Verfahren unter Verwendung dreier CPTs, zusammengesetzte Wörter in deren einzelne Bestandteile zu zerlegen und diese auf deren Grundformen zu reduzieren.[7] Diese drei CPTs werden erstellt wie folgt:

- Ein CPT wird auf Trennstellen trainiert, indem die Wörter von vorne betrachtet werden, z.B. erhält *Hochhäuser* die Klassifizierung *4*.
- Ein CPT wird auf Trennstellen trainiert, indem die Wörter von hinten betrachtet werden, z.B. erhält *Hochhäuser* die Klassifizierung *6*.
- Ein CPT enthält die Regeln zur Grundformreduzierung, z.B. erhält *Hochhäuser* die Klassifizierung *5haus*, was bedeutet, dass die letzten 5 Zeichen gelöscht und *haus* angehängt werden muss, um *Hochhaus* zu erhalten.

Ein Auszug aus den Trainingsdaten für die jeweiligen Bäume findet sich auf Seite 85 in Abschnitt A.3.

Zunächst wird ein Wort in seine Bestandteile zerlegt, bevor diese einzeln reduziert werden; es finden also zunächst

5 [Bus90] beschreibt die Bildung der Wurzel so:

> Man erhält die Wurzel, wenn alle Affixe und Flexive von einer Wortform entfernt werden.

Die Wurzel ist nicht zu verwechseln mit dem Stamm, der gebildet wird, indem alle Suffixe entfernt werden.

6 Ausnahmen bestätigen die Regel: Die *Sitzbank* bzw. *Großbank* flektiert nicht genau so wie *Bank*, da hier die Bedeutung verändert wird. So hat die *Bank* zwei Plurale: *Banken* und *Bänke*, *Sitzbank* bzw. *Großbank* aber jeweils nur einen.

7 Genauer schreiben die Autoren von *compound splitting*, also von der Kompositazerlegung. Ein Kompositum ist ein Wort, das aus mehreren selbstständig vorkommenden Wörtern besteht.

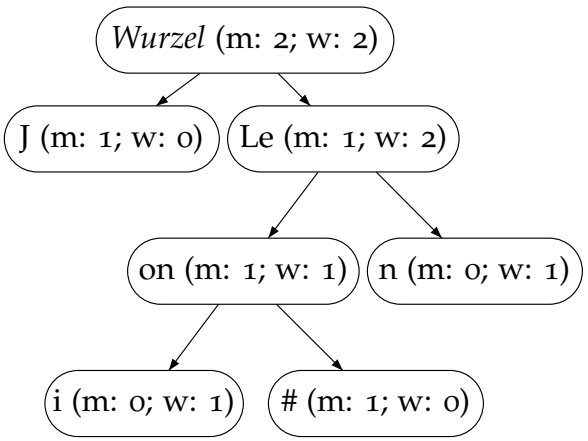

ABBILDUNG 4.3: Beispiel für einen Compact Patricia Trie zur Speicherung von Vornamen mit dem Merkmal *Geschlecht*

nur die ersten beiden CPTs Verwendung. Hierbei können vier Fälle eintreten:

FALL 1 Beide CPTs trennen an der gleichen Stellen – Trennung an dieser Stelle.

FALL 2 Einer der beiden CPTs liefert keine Trennstelle zurück – Trennung an der Stelle des anderen CPTs.

FALL 3 Die Trennstellen stimmen nicht miteinander überein – Trennung an der Stelle mit der höheren Wahrscheinlichkeit.

FALL 4 Beide CPTs liefern keine Trennstelle zurück oder Trennstellen außerhalb der Wortgrenze – keine Trennung.

Das Verfahren wird weiterhin rekursiv auf die einzelnen, bereits getrennten Teilworte angewendet. Der Algorithmus liefert genau eine Zerlegung für ein Wort, obwohl es Worte gibt, für die mehrere Zerlegungen existieren (z.B. *Staubecken* bzw. *Stau-becken* oder *Wach-stube* bzw. *Wachs-tube*).

Silbentrennung

Sowohl die Anzahl der Silben eines Wortes kann Einfluss haben auf die Art der Flexion (z.B. gibt es eine geschlossene Klasse einsilbiger, femininer Substantive, deren Plural auf *e* auslautet), als auch die letzte Silbe eines Wortes: Es gibt z.B. Adjektive wie *frisch*, deren Superlativ mit *-st* oder *-est* gebildet werden kann, da sie auf *sch* enden; der Superlativ von Adjektiven hingegen mit dem Wortbildungssuffix *isch* wie *diebisch* kann nur mit *-st* gebildet werden. Es

ist also eigentlich weniger als die gesamte letzte Silbe ausschlaggebend, aber mehr als nur ein einzelner Buchstabe, der ja bereits erfasst wird. So wurde als Mittelweg für die Implementierung FLEXI die Silbe ausgewählt.

Zunächst muss geklärt werden, was eine **Silbe** ist. Nach [Bus90]:

> Phonetisch-phonologische Grundeinheit des Wortes bzw. der Rede, die zwar intuitiv nachweisbar ist, wissenschaftlich aber nicht einheitlich definiert wird.

Sprachunabhängig besteht eine Silbe aus einem Silbenkern (Nukleus), der meist durch einen oder mehrere Vokale (Diphtong) gebildet wird, davor kann ein Silbenkopf (Onset) stehen, der aus einem oder mehreren Konsonanten gebildet wird, danach kann eine Silbenkoda stehen, die wiederum aus einem oder mehreren Konsonanten besteht. Ausnahmen sind Silbenköpfe wie *s* in dem Ausruf *Pst*.

Da die Silbe der Phonologie zuzuordnen ist, lassen sich die Beschreibungen nicht einfach auf die geschriebene Sprache übertragen, eben auch weil es keine eineindeutige Abbildung zwischen den Elementen der Transkribierung und den Graphemen eines Wortes gibt. Ferner kann auch die Vokallänge ausschlaggebend sein für die Silbentrennung, jedoch ist diese im Deutschen nicht Teil der schriftlichen Repräsentation eines Wortes (vgl. [Hal92]). Trotzdem gibt es nach [KRSSW07] und [fdS06] eindeutige Silbentrennungsregeln für das geschriebene Deutsche.

Um die Anzahl der Silben und die letzte Silbe einer Grundform zu ermitteln, wird *Liangs Algorithmus* ([Lia83]) verwendet. Hierzu wird eine Liste mit Mustern verwendet; die Muster bestehen aus Buchstaben, einem Symbol für Wortanfang und -ende sowie aus Ziffern. Die ungeraden bzw. geraden Zahlen markieren Trenn- bzw. keine Trennstellen, wobei größere Zahlen kleinere überwiegen. Ferner gibt es eine Liste mit Ausnahmen, deren Trennung ansonsten falsch oder ungewollt wäre.[8] Als ungewollt sind solche Trennungen anzusehen, die z.B. den Lesefluss hindern könnten, wie z.B. die Trennung von *Urinstinkt* in *Urinstinkt*.

8 Enthielt die erste Implementierung noch mehrere Tausend Ausnahmen (vgl. [Lia83]), sind es derzeit im Englischen noch weniger als ein Dutzend, siehe `http://tug.ctan.org/tex-archive/language/hyphenation/ukhyphen.tex`.

Die passenden Muster für ein Wort sind diese, deren Buchstabenfolge ein Teil des Wortes sind. So sind die gefundenen Muster bei der genutzten Musterliste[9] für das Wort *Silbenwörter*: *1si*, *2il*, *2lb*, *1be*, *8n1w*, *2r1t*. Diese ergeben zusammengesetzt für das ganze Wort *1S2i2l1boe8n1woö2r1toeoro*, und da nur die ungeraden Zahlen für Trennstellen stehen, ist die Trennung *Sil-ben-wör-ter*.

Da der Algorithmus für das Textsatzsystem TEXentwickelt wurde, gibt es allerdings einige Nachteile im Rahmen der hier beschriebenen Verwendung: Da es nicht gewollt ist (und auch nicht der aktuellen Rechtschreibung entspricht, vgl [fdS06, S. 103], dass am Zeilenende oder -anfang durch Silbentrennung bedingt zu wenige Buchstaben stehen (wie z.B. bei *O-fen*), werden Wortanfänge bzw. -enden und auch sehr kurze Wörter nicht phonologisch korrekt getrennt.

Für FLEXI wird eine eigene Implementierung von *Liangs Algorithmus* verwendet[10].

4.2 ART DER FLEXION

Wie die Merkmale, die für die Flexion bestimmend sind, ermittelt werden können, wurde im Abschnitt 4.1 beschrieben. Nun wird erklärt, welches Verfahren genutzt wird, um automatisch die Art der Flexion zu erkennen.

Im nächsten Schritt ist also ein Algorithmus zu finden, der beschreibt, wie ein Wort flektiert. Hierzu sind die (minimalen) Unterschiede zwischen einer Grundform und einer zugehörigen flektieren Form zu finden, was auf der Zeichenebene zu dem Problem der *Longest Common Subsequence* (LCS) führt (vgl. [HM76]).

4.2.1 *Längste gemeinsame Subsequenz*

Eine Zeichenkette

$$C = c_1 c_2 \dots c_p \tag{4.13}$$

ist nach [Hir75] eine **Subsequenz** von einer Zeichenkette

$$A = a_1 a_2 \dots a_i \tag{4.14}$$

9 Zu finden unter `http://cvs.savannah.gnu.org/viewvc/groff/tmac/hyphen.den?root=groff&view=log`.

10 Zu finden unter `http://bitbucket.org/feuervogel/hyphenation/`.

gdw. eine Abbildung F existiert mit

$$F : \{1, 2, \ldots, p\} \rightarrow \{1, 2, \ldots, i\}, \tag{4.15}$$

so dass $f(j) = k, j \in \{1, \ldots, p\}, k \in \{1, \ldots, i\}$ nur wenn gilt $c_i = a_k$ und wenn F strikt monoton steigend ist. Es gilt $p \leqslant m$.

Eine Zeichenkette C ist eine **gemeinsame Subsequenz** der Zeichenketten A und B gdw. C eine Subsequenz von A und C eine Subsequenz von B ist.

Das Problem der LCS kann nun wie folgt beschrieben werden: Eine **maximale gemeinsame Subsequenz**

$$C = c_1 c_2 \ldots c_p \tag{4.16}$$

zweier Zeichenketten A und B ist eine solche, deren p maximiert ist. Es existiert also keine längere gemeinsame Subsequenz der beiden Zeichenketten A und B.

Die Wahl der längsten gemeinsamen Subsequenz ist jedoch nicht eindeutig: Für die Zeichenketten $A = ab$ und $B = ba$ existieren zwei LCS: $C_1 = a$ und $C_2 = b$.

[Hir75] beschreibt drei Algorithmen zur Ermittlung der LCS, die auf der folgenden Annahme basieren:

$$\mathrm{LCS}(A, B) = \begin{cases} \varepsilon & i = 0 \\ & \vee\ j = 0 \\ (\mathrm{LCS}(A_{i-1}, B_{j-1}), a_i) & a_i = b_j \\ \mathrm{argmax}(\mathrm{LCS}(A_i, B_{j-1}), & a_i \neq b_j \\ \qquad \mathrm{LCS}(A_{i-1}, B_j)) & \end{cases} \tag{4.17}$$

mit $A = a_1 a_2 \ldots a_i, B = b_1 b_2 \ldots b_j$ und ε als Bezeichnung für die leere Zeichenkette. Zu beachten ist: $\mathrm{LCS}(A, B)$ liefert nicht die längste gemeinsame Subsequenz zweier Zeichenketten A und B, sondern eine Menge deren Elemente alle längsten gemeinsamen Subsequenzen sind. Demnach ist $\mathrm{argmax}(P, Q)$ definiert wie folgt:

$$\mathrm{argmax}(P, Q) = \{r \in P \cup Q \mid \nexists s \in P \cup Q : |s| > |r|\} \tag{4.18}$$

Beweis der Korrektheit

GEMEINSAME SUBSEQUENZ Seien

$$A = a_1 a_2 \ldots a_i, B = b_1 b_2 \ldots b_j \tag{4.19}$$

und i beliebig fest.

1. Sei $j = 0 \rightarrow B = \varepsilon$. Die einzige Subsequenz von B ist $C = \varepsilon$, denn nur so ist $F : \emptyset \rightarrow \emptyset$ eine Abbildung nach Definition. Da ε eine Subsequenz jeder Zeichenkette ist, ist ε auch Subsequenz von A und somit gemeinsame Subsequenz von A und B.

2. Sei $a_i = b_j$.

   IA Sei $j = 1$.

$$\begin{aligned} \mathrm{LCS}(A, B) &= \mathrm{LCS}((A_{i-1}, B_0), a_i) \\ &= (\varepsilon, a_i) = a_i = b_j \end{aligned} \tag{4.20}$$

   und da a_i Subsequenz von A und b_j Subsequenz von B ist, gilt unter Voraussetzung $a_i = b_j$: a_i ist gemeinsame Subsequenz von A und B.

   IV Sei $j = x$.

$$\mathrm{LCS}(A, B) = (\mathrm{LCS}(A_{i-1}, B_{j-1}), a_i) \tag{4.21}$$

   IS Sei $j = x + 1$. Ist nun $(\mathrm{LCS}(A_{i-1}, B_x), b_{x+1})$ gemeinsame Subsequenz von A und B? Ja, da nach IV gilt: $\mathrm{LCS}(A_{i-1}, B_x)$ ist Subsequenz von A und B, folgt

$$\exists F : \{1, 2, \ldots, i-1\} \rightarrow \{1, 2, \ldots, x\} \tag{4.22}$$

   und F ist strikt monoton steigend. Da $i > i - 1$ und $x + 1 > x$ ist auch

$$F' : \{1, 2, \ldots, i-1, i\} \rightarrow \{1, 2, \ldots, x, x+1\} \tag{4.23}$$

   monoton steigend.

3. Sei $a_i \neq b_j$.

   IA Sei $j = 1$.

$$\begin{aligned} \mathrm{LCS}(A, B) &= \begin{array}{r} \mathrm{argmax}(\, \mathrm{LCS}(A_{i-1}, B_1), \\ \mathrm{LCS}(A_i, B_0)) \end{array} \\ &= \mathrm{argmax}(\mathrm{LCS}(A_{i-1}, B_1), \varepsilon) \end{aligned} \tag{4.24}$$

   Nun gilt entweder $\exists k \in \{1, 2, \ldots, i-1\}$:

$$a_k = b_1 \rightarrow \mathrm{LCS}(A_{i-1}, B_1) = b_1 \tag{4.25}$$

und daraus folgt

$$\operatorname{argmax}(\mathrm{LCS}(A_{i-1}, B_1), \varepsilon) = b_1 \tag{4.26}$$

oder $\forall k \in \{1, 2, \ldots, i-1\}$:

$$a_k \neq b_1 \rightarrow \mathrm{LCS}(A_{i-1}, B_1) = \varepsilon \tag{4.27}$$

und daraus folgt

$$\operatorname{argmax}(\mathrm{LCS}(A_{i-1}, B_1), \varepsilon) = \varepsilon \tag{4.28}$$

In beiden Fällen ist $\max(\mathrm{LCS}(A_{i-1}, B_1), \varepsilon)$ Subsequenz von A und Subsequenz von B und somit gemeinsame Subsequenz von A und B.

IV Sei $j = x$.

$$\mathrm{LCS}(A, B) = \begin{matrix}\operatorname{argmax}(\ \mathrm{LCS}(A_{i-1}, B_j),\\ \qquad\qquad \mathrm{LCS}(A_i, B_{j-1}))\end{matrix} \tag{4.29}$$

IS Sei $j = x + 1$. Zu überprüfen ist nun

$$\mathrm{LCS}(A, B) = \begin{matrix}\operatorname{argmax}(\ \mathrm{LCS}(A_{i-1}, B_{x+1}),\\ \qquad\qquad \mathrm{LCS}(A_i, B_x))\end{matrix} \tag{4.30}$$

In jedem Fall ist $s_2 = \mathrm{LCS}(A_i, B_x)$ nach IV eine Subsequenz von A und B. Ist auch $s_1 = \mathrm{LCS}(A_{i-1}, B_{x+1})$ eine Subsequenz so wird die längere von beiden gewählt (sind beide gleich lang, beide), ist s_1 keine Subsequenz, so ist s_2 wie bereits bewiesen Subsequenz.

MAXIMALITÄT Sei

$$A = a_1 a_2 \ldots a_i, B = b_1 b_2 \ldots b_j \tag{4.31}$$

und i beliebig fest.

1. Sei $j = 0$. Es folgt direkt aus der Definition der Subsequenz, dass ε die längste Subsequenz von B und eine Subsequenz von A ist und somit die längste gemeinsame Subsequenz von A und B.

2. Sei $a_i = b_j$. Es ist schnell ersichtlich, dass $(\mathrm{LCS}(A_{i-1}, B_{j-1}), a_i)$ maximal ist:

IA Sei $j = 1$.

$$a_i = b_j \rightarrow \mathrm{LCS}(A, B) = b_1 \tag{4.32}$$

Es folgt wiederum direkt aus der Definition der Subsequenz, dass b_1 die maximale Subsequenz von A und B ist.

IV Sei $j = x$.

$$\mathrm{LCS}(A, B) = \mathrm{LCS}(\mathrm{LCS}(A_{i-1}, B_{j-1}), a_i) \tag{4.33}$$

IS Sei $j = x + 1$. Sei nun

$$\begin{aligned} a_i = b_{x+1} &\rightarrow \mathrm{LCS}(A, B) \\ &= \mathrm{LCS}(\mathrm{LCS}(A_{i-1}, B_x), b_{x+1}) \end{aligned} \tag{4.34}$$

falsch, also es existiert

$$\begin{aligned} D &= d_1 d_2 \dots d_{y-1} d_y \\ &= \mathrm{LCS}(A, B), y > x + 1 \end{aligned} \tag{4.35}$$

und daraus folgt

$$\begin{aligned} D_{y-1} &= d_1 d_2 \dots d_{y-1} \\ &= \mathrm{LCS}(A_{i-1}, B_x), y - 1 > x \end{aligned} \tag{4.36}$$

Dies steht aber im direkten Widerspruch zur Induktionsvoraussetzung, da

$$\mathrm{LCS}(A_{i-1}, B_x) = c_1 c_2 \dots c_x \tag{4.37}$$

3. Sei $a_i \neq a_j$.

IA Sei $j = 1$. Die im Beweis zur gemeinsamen Subsequenz gefundenen möglichen Subsequenzen sind jeweils maximal: entweder b_1 ist Subsequenz von A, dann ist b_1 zugleich maximale gemeinsame Subsequenz von B, oder b_1 ist nicht Subsequenz von A, dann ist ε die längste gemeinsame Subsequenz.

IV Sei $j = x$.

$$\mathrm{LCS}(A, B) = \begin{matrix} \mathrm{argmax}(\mathrm{LCS}(A_{i-1}, B_x), \\ \mathrm{LCS}(A_i, B_{x-1})) \end{matrix} \tag{4.38}$$

IS Sei $j = x + 1$. Ist nun

$$\begin{aligned} C = c_1 c_2 \dots c_p &= LCS(A, B) \\ &= \begin{array}{l} \text{argmax}(\, LCS(A_{i-1}, B_{x+1}), \\ \qquad\qquad LCS(A_i, B_x)) \end{array} \end{aligned} \tag{4.39}$$

maximale Subsequenz? Es sind drei Fälle zu unterscheiden:

a) Sei $c_p = a_i, c_p \neq b_{x+1}$. Nun ist die längste gemeinsame Subsequenz von A und B folglich $LCS(A_i, B_x)$ und diese ist nach IV maximal.
b) Sei $c_p = b_{x+1}, c_p \neq a_i$. Nun ist die längste gemeinsame Subsequenz von A und B folglich $(LCS(A_{i-1}, B_x), b_{x+1})$. (Da $LCS(A_{i-1}, B_x)$ bereits längste gemeinsame Subsequenz von A_{i-1} und B_x ist, ist die um das Element b_{x+1} erweiterte längste gemeinsame Subsequenz A und B wiederum maximal.)
c) Sei $c_p \neq a_i, c_p \neq b_{x+1}$. So ist die längste gemeinsame Subsequenz von A und B folglich $LCS(A_{i-1}, B_x)$ und diese ist nach IV bereits maximal.

Die Beweise zu j beliebig fest sind analog. □

Beispiel

Sei A = Haus und B = Häuser. Durch den Algorithmus wird eine Matrix der Größe $(|A| + 1) * (|B| + 1)$ erzeugt (siehe Tabelle 4.1 auf Seite 48).

		H	ä	u	s	e	r
	ε	ε	ε	ε	ε	ε	ε
H	ε	H	H	H	H	H	H
a	ε	H	H	H	H	H	H
u	ε	H	H	Hu	Hu	Hu	Hu
s	ε	H	H	Hu	Hu	Hus	**Hus**

TABELLE 4.1: Die längste gemeinsame Subsequenz am Beispiel *Haus – Häuser*

4.2.2 *Unterschiede zwischen Wörtern*

Um zu beschreiben, wie ein Wort flektiert, müssen aber nun nicht die Gemeinsamkeiten zwischen der Grundform und der flektierten Form betrachtet werden, sondern die Veränderungen, die von der Grundform zu der flektierten Form führen. Hierzu wird *diff* verwendet (vgl. [HM76]), dessen Kern aus einem Algorithmus zum Erstellen der LCS besteht; daher wird an dieser Stelle auf eine detaillierte Erklärung verzichtet.

Ursprünglich wurde *diff* als Unix-Kommandozeilen-Werkzeug `diff` implementiert, um die Unterschiede zweier Dateien *file_a* und *file_b* (verglichen werden jeweils die Zeilen) darzustellen. Die Ausgabe kann als sogenannter *Patch* gespeichert werden und mittels `patch` auf *file_a* angewendet werden, um *file_b* zu erzeugen.

```
1  :~$ cat file_a
2  a
3  b
4  c
5  :~$ cat file_b
6  b
7  d
8  c
9  e
10 :~$ diff file_a file_b
11 1d0
12 < a
13 2a2
14 > d
15 3a4
16 > e
```

Das Programm `diff` erzeugt das gezeigte Patch, in welchem folgende Befehle enthalten sind: Lösche *a* in Zeile 1, füge in Zeile 2 *d* und an Zeile 3 *e* an. Die Änderungsbefehle beschreiben dreierlei: in welchen Zeilen eine wie Datei geändert, in welchen Zeilen aus einer Datei was gelöscht und in welchen Zeilen in eine Datei was eingefügt werden soll.

Für den Einsatz in Flexi sollten allerdings folgende Wortveränderungen beherrscht werden: Voranstellen (z.B. von Präfixen), Ändern (wie z.B. bei Ablautung oder Umlautung) und Anhängen (wie z.B. von Suffixen), daher wurde eine Eigenimplementierung realisiert. Das Einfügen (wie z.B. von Infixen) wird durch das Voranstellen innerhalb des Wortes realisiert.

Zudem wäre es nunmehr von Vorteil, wenn die Bezugnahme der Änderungsbefehle nicht mit einer Nummerierung von links nach rechts geschähe.

4.2.3 *Kennzeichnung von Buchstaben*

Der naivste Ansatz, die Zeichen eines Wortes von links nach rechts durch Nummern zu kennzeichnen, ist zugleich der denkbar ungünstigste: *Uhu* und *Auto* flektieren analog; nummeriert man jedoch von links nach rechts, wird das *s* für den Genitiv Singular bzw. für den Plural nach dem dritten (*Uhu*) respektive nach dem vierten (*Auto*) Buchstaben angehängt; es lassen sich also die Veränderungen von *Uhu* nach *Uhus* nicht auf *Auto* übertragen, denn fügt man das *s* bei *Auto* nach der dritten Stelle (nämlich wie bei *Uhu*) an, ensteht **Autso*.

Kennzeichnung von rechts nach links

Um diese Problematik zu umgehen, ist es besser die umgekehrte Nummerierung von rechts nach links zu verwenden. So können Wörter *Auto* und *Uhu* nach dem gleichen Schema flektiert werden: hinter den am weitesten rechts stehenden Buchstaben wird das *s* angefügt. Gleiches würde man auch erreichen, wenn man das Wortende als solches kennzeichnen würde. Da bei der Flexion jedoch nicht nur die Anfügung von Suffixen vorgenommen wird, sondern beispielsweise auch Umlautung, ist dieses Vorgehen nicht zielführend.

Bei einer Nummerierung von rechts nach links haben Wörter, die den gleichen umlautenden Vokal an gleicher Stelle (von rechts betrachtet) stehen haben dann auch das gleiche Schema: *Blatt* und *Land* bspw. haben den Wechsel von *a* nach *ä* von rechts betrachtet an der gleichen Stelle (aber nicht von links betrachtet) und somit das gleiche Schema.

Weitere Probleme ergeben im Zusammenhang mit der Um- bzw. Ablautung sowie der Verwendung von In- bzw. Präfixen: Wörter, die den umlautenden Vokal an unterschiedlichen Stellen haben, flektieren nach unterschiedlichen Schemata; ferner ist das Einfügen eines Infixes (bzw. Präfixes) vor den Stamm (wie z.B. bei *um**ge**fahren*) aus der Sicht des Linguisten nicht korrekt „nachgeahmt“: Denn

hier wird nicht an der n. Stelle von links bzw. rechts eingefügt, sondern vor der Wurzel.

Morphemkennzeichnung

Eine Lösung für die Probleme durch die reine Buchstabenkennzeichnung ist eine zusätzliche Kennzeichnung der Morphemgrenzen (also der Morphemanfänge bzw. der -enden). Der Algorithmus hierfür ist bereits bekannt und implementiert (siehe Abschnitt 4.1.1 auf Seite 40). Auch hier erfolgt eine Markierung so, dass die Wurzelgrenzen bei Wörtern mit unterschiedlicher Anzahl an Morphemen aber gleicher Wurzel gleich markiert werden (also wiederum von rechts nach links, da die Trainingsdaten so generiert wurden, dass die Wurzel das am weitesten rechts stehende Morphem ist[11]).

So kann nun auch noch der **Stammvokal** zur Umlautung markiert werden. Nach [Bus90] wird der Stammvokal definiert als der Vokal, der um- bzw. ablautet. Doch diese Definition ist trifft nicht zu, wenn man bei einem Wort, dessen um- bzw. ablautenden Vokal nicht kennt, genau diesen markieren will. Außerdem würden demnach Wörter, die nicht um- bzw. ablauten, keinen Stammvokal besitzen. Pragmatischerweise wird hierzu der am weitesten links stehende Vokal der Wurzel verwendet.

Das Wort *Hochhaus* wird also korrekterweise wie folgt markiert:

Buchstaben	H	o	c	h	h	a	u	s
Morpheme	2. Anfang			2. Ende	1. Anfang	Stammvokal		1. Ende
Rechts nach links	8	7	6	5	4	3	2	1

Tabelle 4.2: Kennzeichnung am Beipiel *Hochhaus*

Gleiches gilt analog für Verben (z.B. für das Einfügen des Präfixes *ge* vor *entgegen**ge**fahren* oder das Ablauten des *a* in *entgegenf**u**hr*) und für die Steigerung von Adjektiven bzw. Adverben.

11 Verben wie z.B. *hinabgehen* werden getrennt in *hinab* und *gehen*, d.h. auf eine Abtrennung des Suffix-Morphems wird explizit verzichtet; anderenfalls ist eine wortartunabhänige Markierung der Wurzel nicht möglich.

4.3 GENERIERUNG DER WORTFORMEN

Als Grundlage für die Generierung von Wortfromen stehen nun Grundformen zur Verfügung, denen flektierte Formen und für die Flexion ausschlaggebende Merkmale zugeordnet werden können. Ferner kann vermerkt werden, welche Kombinationen der morphosyntaktischen Merkmale durch die flektierten Formen möglich sind. Diese Datensammlung wird im Folgenden als **Lexikon** bezeichnet.

4.3.1 *Entscheidungsfindung*

Zur Entscheidung, wie eine beliebige Grundform (des Deutschen) flektiert wird, sind im Groben folgende Schritte zu befolgen:

1. Befindet sich die Grundform bereits im Lexikon, werden deren flektierte Formen zusammen mit den morphosyntaktischen Merkmalen zurückgegeben.
2. Befindet sich die Grundform nicht im Lexikon, wird versucht, die für die Flexion ausschlaggebenden Merkmale herauszufinden, um daraufhin die flektierten Formen zu erzeugen und zurückzugeben.

Schritt 1 ist wichtig, um Ausnahmen beachten zu können. So können z.B. Adjektive eingefügt werden, die auf *el* enden, deren *e* der Endung aber durch die Flexion nicht wegfällt (z.B. *eitel* und *eitler Mann* aber *fidel* aber **fidler Mann*. Die unterschiedliche Flexion von Ajdektiven auf *el* wird mit der unterschiedlichen Betonung begründet, die nur sehr schwer – wenn überhaupt – automatisch zu Erfassen ist). Dieser Schritt wird nicht näher beschrieben; die Struktur des Lexikons muss lediglich ermöglichen, darin nach Grundformen zu suchen.

Nun zu Schritt 2. Es wurde bereits beschrieben wie die Merkmale auf Basis einer einzelnen Grundform, die quasi kontext-frei, also nicht im Gefüge eines oder mehrere Satzes betrachtet wurde (siehe Abschnitt 4.1.1). Ferner wurde erläurtert, wie sich die Veränderungen durch Morphemkennzeichnungen so abstrahieren lassen, dass sie nicht mehr an ein konkretes Wort gebunden sind, sondern sich auf die eingegebene Grundform übertragen lassen (siehe Abschnitt 4.2).

Um von einer Menge von Merkmalsausprägung auf die Art der Flexion zu schließen, werden im Folgenden zwei

verschiedene Verfahren zur Klassifikation vorgestellt: der *Entscheidungsbaum* und der *Naive Bayes'sche Klassifizierer*.

Entscheidungsbaum (nach [RN03])

Ein Entscheidungsbaum ist einer der einfachsten, zudem aber auch einer der besten Algorithmen des (maschinellen) Lernens; er trifft auf der Basis einer Menge von Merkmalsausprägungen, die ein Objekt oder eine Situation beschreiben, eine „Entscheidung" – einen Rückgabewert für die Eingabe. Sowohl die Eingabe – als auch die Ausgabewerte können diskret wie auch kontinuierlich sein.

Eine Entscheidung wird durch eine Reihe von Abfragen getroffen. Die inneren Knoten des Entscheidungsbaums repräsentieren jeweils eine Ausprägung eines Merkmals. Die Blätter hingegen sind versehen mit den möglichen Rückgabewerten. Hat man nun eine Menge von Merkmalsausprägungen muss nur der Entscheidungsbaum bis zum entsprechenden Blatt „abgeschritten" werden.

BEISPIEL Die Funktionsweise eines Entscheidungsbaumes lässt sich gut an einem Beispiel erklären, da auch (rationale) menschliche Entscheidungen meist ähnlich getroffen werden.

Im folgenden sei ein Entscheidungsbaum beschrieben, der die Wohnungssuche in Leipzig erleichtern soll. Dazu seien folgende Testdaten gegeben:

Merkmale	Balkon	Stadtteil	Bad mit Tageslicht	Badewanne	Kommt in Frage
Wohnung 1	Nein	Lindenau	Ja	Nein	Nein
Wohnung 2	Nein	Plagwitz	Nein	Ja	Ja
Wohnung 3	Nein	Plagwitz	Ja	Nein	Nein
Wohnung 4	Nein	Lindenau	Ja	Ja	Ja
Wohnung 5	Ja	Grünau	Ja	Nein	Ja
Wohnung 6	Nein	Gohlis	Ja	Nein	Nein
Wohnung 7	Ja	Grünau	Ja	Ja	Nein

TABELLE 4.3: Trainingsdaten für einen Entscheidungsbaum

Ein Entscheidungsbaum, der sich aus diesen Daten bilden lässt, ist zu sehen in Abbildung 4.4 auf Seite 54. Hier ist zudem eine Eigenart zu erkennen: Der Entscheidungsbaum entsteht durch die Beispieldaten und bildet nicht die korrekte Funktion der Entscheidungsfindung ab; mag bei

der Wohnungssuche (und beim Erstellen der Beispieldaten) der Stadtteil eine Rolle spielen, tut er das nicht mehr in dem abgebildeten Entscheidungsbaum in Abbildung 4.4 auf Seite 54.

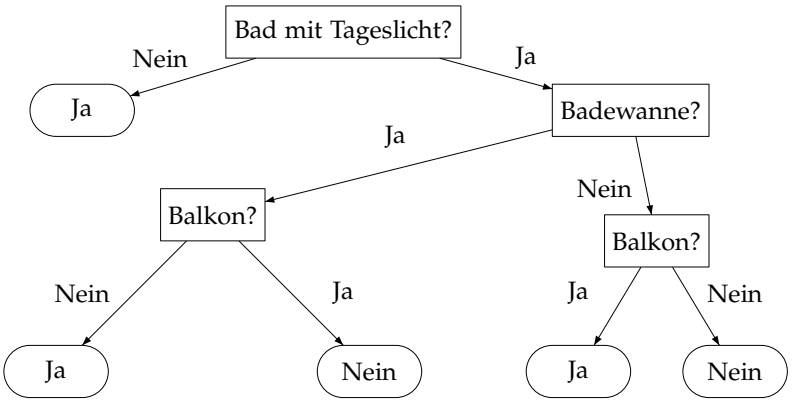

ABBILDUNG 4.4: Beispiel für einen Entscheidungsbaum

ERSTELLEN EINES ENTSCHEIDUNGSBAUMS Es mag zunächst schwierig erscheinen, aus gegebenen Daten einen Entscheidungsbaum zu erstellen, es existiert jedoch eine triviale Lösung: Für jeden Datensatz aus den Beispieldaten wird ein Pfad von der Wurzel zu einem Blatt entlang der Merkmalsausprägungen gebildet. Hieraus lassen sich jedoch keinerlei Muster ableiten, so dass sich für unbekannte Eingabedaten keine Voraussagen treffen lassen. Nach dem Prinzip von *Ockhams Messer*[12] sollte jedoch der *kleinste* Entscheidungsbaum zu den Beispieldaten gefunden werden; da dies jedoch (für die unterschiedlichen Definitionen für *klein*) als unlösbar gilt, reicht es aus, einen *kleinen* Baum zu finden.

Der **Decision-Tree-Learning** Algorithmus (vgl. [RN03, S. 685]) wählt aus der Menge der Merkmale zunächst das wichtigste Merkmal aus; das wichtigste ist jenes, das den größten Unterschied in der Klassifikation macht. Mit den

12 *Ockhams Messer* (engl. *Occam's Razor* oder auch *Ocham's Razor*) meint das Prinzip unter mehreren gleichwertigen Erklärungen die einfachste zu bevorzugen. Im Fall des Entscheidungsbaumes besagt das Prinzip, dass nicht mehr Merkmale zu verwenden sind, also unbedingt nötig, um die Trainingsmenge möglichst korrekt zu beschreiben. Hat man zwei Entscheidungsbäume, die exakt die gleichen Annahmen treffen, ist der einfachere zu bevorzugen.

übrig gebliebenen Merkmalen wird nun rekursiv weiterverfahren, denn was folgt, ist wiederum ein Entscheidungsbaum. Ziel ist es, möglichst kurze Pfade durch möglichst gute Entscheidungskriterien zu erhalten, wodurch ein kleiner Baum entsteht.

In dem Algorithmus wird eine Funktion zum Auswählen des nächsten Attributes verwendet: die **Choose-Attribute** Funktion. Diese hat die Eigenschaft, dass sie das Attribut zurückgibt, das die meiste **Information**[13] liefert: je weniger über den Wert eines Merkmals vorausgesagt werden kann, desto mehr Information liefert dieser. Je höher also der Wert einer Funktion zur Berechnung der Information eines Merkmals, desto höher die Information des Merkmals. Angenommen die Möglichen Werte eines Merkmals $v_1, \ldots, v_n$ treten mit der Wahrscheinlichkeit $P(v_i), i \in \{1, \ldots, n\}$ auf, dann ist die Funktion zur Informationsberechnung eines Merkmals I wie folgt definiert:

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) * \log_2(P(v_i)) \tag{4.40}$$

Will man also den Wert eines Ereignisses wie dem Münzwurf errechnen und nimmt man an, dass beide Seiten gleich wahrscheinlich eintreten können, so berechnet sich die Information wie folgt:

$$\begin{aligned} I(0.5, 0.5) &= -0.5 * \log_2(0.5) - 0.5 * \log_2(0.5) \\ &= 1 \end{aligned} \tag{4.41}$$

Wenn die Wahrscheinlichkeit Kopf zu treffen bei 0.99 läge, wäre die Information wie folgt:

$$\begin{aligned} I(0.99, 0.01) &= -0.99 * \log_2(0.99) - 0.01 * \log_2(0.01) \\ &= 0.08 \end{aligned} \tag{4.42}$$

Intuitiv ist das klar: Wenn es sehr wahrscheinlich ist, dass Kopf getroffen wird, ist die Information des Münzwurfs nicht hoch wie im Fall des fairen Münzwurfs.

Für das Beispiel zur Wohnungsfindung sind folgende Werte zu berechnen:

13 Der Begriff der Information wird in dem Sinn von [Wea49] verwendet.

BALKON $I(\frac{5}{7}, \frac{2}{7}) = 0.86$

STADTTEIL $I(\frac{2}{7}, \frac{2}{7}, \frac{2}{7}, \frac{1}{7}) = 1.950$

BAD MIT TAGESLICHT $I(\frac{1}{7}, \frac{6}{7}) = 0.592$

BADEWANNE $I(\frac{3}{7}, \frac{4}{7}) = 0.985$

Somit ergibt sich der neue, aber gleichwertige Entscheidungsbaum in Abbildung 4.5 auf Seite 56. Wie zu sehen ist, ist die maximale Pfadlänge nun nicht mehr drei, was daran liegt, dass nur noch zwei Merkmale (und nicht mehr drei) eine Rolle spielen.

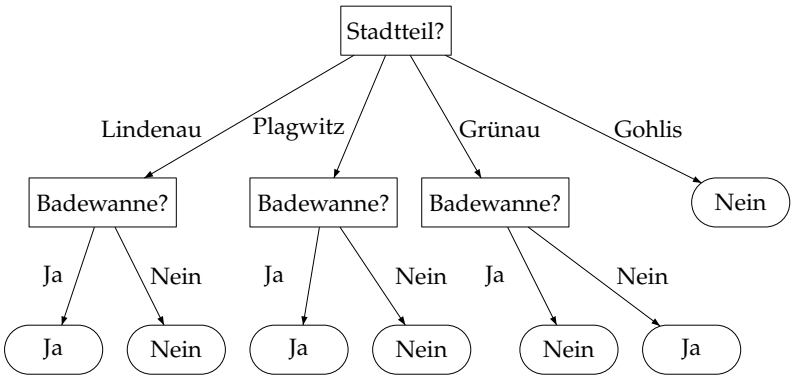

ABBILDUNG 4.5: Beispiel für einen kleinen Entscheidungsbaum

NOISE Sind die Daten inkonsistent, d.h. es gibt für die gleichen Merkmalsausprägungen unterschiedliche Klassifikationen, spricht man von **Noise**. Der Entscheidungsbaum kann dann nicht die gesamte Trainingsmenge nachbilden. Eine Möglichkeit ist hier, die Klassifikation nach der mehrheitlich verwendeten Merkmalsausprägung zu bestimmen. Alternativ können auch alle Klassifikationen mit ihren relativen Häufigkeiten zurückgegeben werden.

OVERFITTING Ist eine aus Trainingsdaten abgeleitete Hypothese zwar bezogen auf die Trainingsdaten sehr gut, spiegelt sie aber nicht die Realität wieder, spricht man von **Overfitting**. Ein Weg dies zu vermeiden ist das **Pruning**: Irrelevante Merkmale (also solche, mit einem geringen Informations-Wert) werden weggelassen.

IMPLEMENTIERUNG Es wird das Weka-Paket[14] genutzt. Dieses verwendet u.a. den C4.5-Algorithmus (vgl. [Qui93])

14 Zu finden unter `http://www.cs.waikato.ac.nz/~ml/`

zum Erstellen des Entscheidungsbaums, der sowohl mit fehlenden Werten eines Merkmals als auch mit diskreten und kontinuierlichen Werten umgehen kann. Alternativ kann auch dessen Vorgänger, der ID3-Algorithmus (vgl. [Qui86]), eingesetzt werden.

Naiver Bayes'scher Klassifizierer (nach [Nil98])

Der Naive Bayes'scher Klassifizierer ist ein probalistischer Klassifizierer, d.h. er kann für die Merkmale $M_1, \ldots, M_i$ die Wahrscheinlichkeit berechnen, dass eine Klassifizierung C eintritt: $P(C \mid M_1, \ldots, M_i)$.

Dieser Klassifizierer basiert auf dem Bayes'schem Theorem:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{4.43}$$

Umformuliert lautet die Gleichung für die Wahrscheinlichkeit der Klassifizierung C unter der Bedingung $M_1, \ldots, M_i$ wie folgt:

$$P(C \mid M_1, \ldots, M_i) = \frac{P(M_1, \ldots, M_i \mid C)P(C)}{P(M_1, \ldots, M_i)} \tag{4.44}$$

Für jede mögliche Ausprägung von C ist der Nenner des Bruchs konstant, es variiert lediglich der Zähler, der sich mit auch mit Mitteln der Verbundwahrscheinlichkeit beschreiben lässt:

$$P(M_1, \ldots, M_i \mid C)P(C) = P(M_1, \ldots, M_i, C) \tag{4.45}$$

Der *Naive* Bayes'sche Klassifizierer basiert des weiteren auf der *naiven* Annahme, dass die Merkmale voneinander stochastisch unabhängig auftreten, d.h. es gilt

$$\forall j, k : P(M_j \mid C, M_k) = P(M_j \mid C), j \neq k \tag{4.46}$$

.

Daraus ergibt sich die verwendete Gleichung:

$$P(M_1, \ldots, M_i \mid C) = \prod_{j=1}^{i} P(M_j \mid C) \tag{4.47}$$

Ist nun für eine gegebene Ausprägung dieser Merkmale die passende Klassifizierung zu berechnen, wird jene ausgewählt, für die $\prod_{j=1}^{i} P(M_j|C)$ maximal ist.

Wird der Naive Bayes'sche Klassifizierer nun mit den Trainingsdaten aus Tabelle 4.3 auf Seite 53 trainiert, ergibt sich z.B. für die Klassifizierung in *Ja* und *Nein* und die Merkmale *Balkon: Nein*, *Stadtteil: Plagwitz*, *Bad mit Tageslicht: Nein* und *Badewanne: Nein* folgende Werte:

$$\begin{aligned} &P(\text{Nein}, \text{Plagwitz}, \text{Nein}, \text{Nein} \mid \text{Nein}) \\ &= \frac{3}{4} * \frac{1}{4} * \frac{0}{4} * \frac{3}{4} = 0 \end{aligned} \tag{4.48}$$

und

$$\begin{aligned} &P(\text{Nein}, \text{Plagwitz}, \text{Nein}, \text{Nein} \mid \text{Ja}) \\ &= \frac{2}{3} * \frac{1}{3} * \frac{1}{3} * \frac{1}{3} = \frac{2}{81} \end{aligned} \tag{4.49}$$

Für die genannten Merkmale wird also die Klassifizierung *Ja* gewählt.

Fazit und Vergleich beider Verfahren

Bei einem Entscheidungsbaum ist die Klassifizierung recht einfach, sehr komplex dagegen ist das Erstellen. Der Entscheidungsbaum ist nicht erweiterbar, d.h. es müssen für das Erstellen sämtliche Datensätze vorliegen. Dies kann bei einer großen Anzahl an Datensätzen schnell zu Komplexitätsproblemen führen. Hier ist also zu entscheiden, wie die Daten so reduziert werden können, dass darunter nicht die Qualität leidet. Zudem spiegelt der Entscheidungsbaum nicht die Realität wider, sondern stellt ledlich die Trainingsdaten dar. Es kann also sein, dass bisher als wichtig erachtete Merkmale (wie z.B. die Wortart) gänzlich weggelassen werden.

Bei dem Naiven Bayes'schen Klassifizierer dagegen müssen nicht alle Trainingsdaten auf einmal gelernt werden, da die eigentliche Berechnung erst bei der Klassifizierung statt findet. So kann eventuell mit mehr Trainingsdaten gelernt werden, auch werden Merkmale nicht einfach weggelassen, sondern sämtliche Merkmale in die Berechnung der Wahrscheinlichkeit mit einbezogen. Es bleibt zu ermitteln,

ob diese Eigenschaften auch zu einer gesteigerten Qualität führen.

4.4 GESAMT-ÜBERSICHT AM BEISPIEL

Bisher wurden in diesem Kapitel die bereits in Verwendung befindlichen Methoden vorgestellt, die Anwendung FLEXI entwickelt und ihr Ablauf skizziert. Im Folgenden soll eine Übersicht über den gesamten Ablauf von FLEXI anhand eines Beispiels gegeben werden.

Als Beispiel dient das Adverb *oft*. Es ist sehr kurz und hat nur drei Formen: das Positiv *oft*, den Komparativ *öfter* und den Superlativ *oftest*.

4.4.1 *Erstellen des Lexikons*

Die Ausgangsdaten für das Erstellen des Lexikons bestehen aus der Grundform, einer flektierten Form und einem Bezeichner für die morphosyntaktischen Merkmale. Für *oft* sehen diese Daten also wie folgt aus:

```
1 oft     oft     Pos
2 oft     öfter    Komp
3 oft     öftest   Sup
```

Nun wird die Lexikon-Datei erstellt, in dem die Merkmale ermittelt werden, die für die Flexion bestimmend sind, die Kennzeichnungen vorgenommen werden, um später die Art der Flexion einfacher übertragen zu können und für jede flektierte Wortform die zugehörigen morphosyntaktischen Merkmalskombinationen zugeordnet werden.

Für das Beispiel *oft* wird also konkret folgendes persistent gespeichert:

GRUNDFORM oft

MERKMALE

- *de_prefix*: Der Präfix ist leer.
- *de_pos*: Wortart ist *Adverb*.
- *de_stem_vowel*: Stammvokal ist *o*.
- *de_word*: Das Wort selbst ist *oft*.
- *de_gender*: Das Geschlecht ist *Neutrum*.[15]
- *de_being*: Das Adverb *oft* ist selbsterklärend kein Lebewesen.

15 Diese Angabe ist an dieser Stelle eigentlich nicht sinnvoll, wird aber für jede Grundform unabhängig von der Wortart ermittelt.

- *de_end_of_word*: Das Wort endet mit *t*.
- *de_hyphen_count*: Die Anzahl der Silben ist 1.
- *language*: Als Sprache wird bei dem Erstellen des Lexikons für das Deutsche ausschließlich Deutsch ausgewählt.[16]
- *de_root*: Wurzel ist *oft*.
- *de_last_hyphen*: Letzte Silbe ist *oft*.
- *de_abbr*: Es handelt sich nicht um eine Abkürzung.

KENNZEICHNUNGEN

- *Position 0 (o)*: Stammvokal.
- *Position 1 (f)*: zweites Zeichen von rechts.
- *Position 2 (t)*: Ende des ersten Morphems, gezählt von rechts.

FLEKTIERTE FORMEN

- *oft* drückt den Positiv aus.
- *öfter* drückt den Komparativ aus.
- *öftest* drückt den Superlativ aus.

4.4.2 *Erstellen des Klassifizierers*

Die Knoten des Entscheidungsbaums bzw. die Merkmale des Naive Bayes'schen Klassifizierers werden durch die Merkmale wie Wortart, Geschlecht, Anzahl der Silben usw. gebildet.

Die Klassifizierungen sind jedoch eine nähere Erläuterung wert. Es wird zur Laufzeit ermittelt, wie sich die einzelnen Grundformen verändern. Zusätzlich zu der jeweiligen Veränderung werden die zugehörigen Bezeichner für die morphosyntaktischen Merkmale gespeichert.

Für *oft* wird also folgendes ermittelt:

1. Verändere nichts. Dies repräsentiert den Positiv.
2. Ersetze das Zeichen mit der Kennzeichnung *stem_vowel* durch *ö* und hänge an das Zeichen mit der Kennzeichnung *end_morph_1* die Zeichen *er* an. Dies repräsentiert den Komparativ.
3. Ersetze das Zeichen mit der Kennzeichnung *stem_vowel* durch *ö* und hänge an das Zeichen mit der Kennzeichnung *end_morph_1* die Zeichen *est* an. Dies repräsentiert den Superlativ.

16 So ist es möglich, dass in dem Lexikon Wörter unterschiedlicher Sprachen enthalten sein können. Die Sprache eines Wortes wird somit zu einem weiteren Merkmal reduziert.

Zusätzlich dazu wird ein eindeutiger Bezeichner für diese Art der Flexion generiert, der als Klassifizierung dient. Wird bei der Generierung flektierter Formen dieser Bezeichner als Klassifizierung für ein anderes Wort ermittelt, können die Veränderungen samt dem Bezeichner für die morphosyntaktischen Merkmale angewandt werden.

5

EVALUATION

Nachdem im letzten Kapitel sowohl die verwendeten Methoden und Verfahren als auch der gesamte Algorithmus für FLEXI beschrieben wurden, werden sie in diesem Kapitel evaluiert. Nur so lässt sich objektiv beschreiben, ob die Beobachtungen aus Kapitel 2 zu den richtigen Schlussfolgerungen geführt haben. Hierfür werden zunächst geeignete Evaluationsmaße vorgestellt. Danach folgen die Messungen der einzelnen Verfahren und schließlich die Berechnung der Güte von FLEXI.

5.1 EVALUATIONSMASSE (NACH [RN03])

Für die Evaluation von Systemen des *Information Retrieval*[1] werden traditionell zwei Maße verwendet: **Precision** (zu deutsch *Grad der Genauigkeit*) und **Recall** (zu deutsch *Grad der Erschöpfung*). Soll z.B. ein System zur Klassifizierung von Dokumenten in relevante und nicht relevante Dokument dahingehend evaluiert werden, wie gut es relevante Dokumente findet, ergeben sich aus der Menge der Dokumente $\cup$ vier disjunkte Teilmengen: A beschreibt die relevanten gefundenen Dokumente, B die nicht relevanten gefundenen Dokumente, C die relevanten nicht gefundenen Dokumente und D die nicht relevanten nicht gefundenen Dokumente (siehe auch Tabelle 5.1 auf Seite 63).

$\cap$	Relevant	Nicht Relevant
Gefunden	A	B
Nicht Gefunden	C	D

TABELLE 5.1: Ausgangsmengen zur Berechnung von Precision und Recall

1 *Information Retrieval* ist ein Teilgebiet der automatischen Sprachverarbeitung, dass sich mit dem Finden von Dokumenten, die für den Nutzer relevant sind, beschäftigt.

Die Qualität der Menge der gefundenen Dokumente $A \cup B$ wird durch die Precision gemessen:

$$p := \frac{|A|}{|A \cup B|} \tag{5.1}$$

Der Grad der Erschöpfung, wie viele der relevanten Dokumente gefunden wurden, wird durch den Recall gemessen:

$$r := \frac{|A|}{|A \cup C|} \tag{5.2}$$

Als abgeleitetes Maß, das aus den beiden Maßen Precision und Recall kombiniert wird, existiert der **F-Wert** (aus dem Englischen *F-Value*, *F-Score* oder *F-Measure*). Der F-Wert ist das harmonische Mittel[2] aus Precision p und Recall r:

$$f := \frac{2 * p * r}{p + r} \tag{5.3}$$

Gilt $p = r = 0$ so wird üblicherweise festgelegt, dass $f = 0$ gilt, da der F-Wert ansonsten undefiniert bliebe. Anzustreben sind für alle drei Maße möglichst hohe Werte, da dies dem Verfahren eine hohe Qualität bescheinigt (vgl. [HQW06, S. 256]).

Die eingeführten Maße dürfen aber nicht als Güte für das gemessene Verfahren allein betrachtet werden. Es ist zudem wichtig zu berücksichtigen, wie die Referenzdaten ermittelt wurden und, falls das Verfahren Wissen aus einer Trainingsmenge bezieht, welche Trainingsmenge verwendet wurde.

5.2 EVALUATION DER ANGEWANDTEN VERFAHREN

Im folgenden werden nun die eingesetzten Verfahren im Einzelnen ausführlich evaluiert. Versagt eines der eingesetzten Verfahren gänzlich, kann dies dazu führen, dass dadurch die Qualität von FLEXI beeinträchtigt wird.

5.2.1 *Geschlechtsbestimmung von Substantiven*

Es soll das Verfahren aus Abschnitt 4.1.1 auf Seite 39 evaluiert werden, für das ein Compact Patricia Trie eingesetzt wird.

2 Auf die gewichtete Variante wurde verzichtet, da es keinen Grund gibt, Precision oder Recall als wichtiger einzustufen.

Wird für die Substantive aus der Trainingsmenge (Ausschnitt siehe Abschnitt A.1 auf Seite 83) das Geschlecht ermittelt, ergeben sich folgende Werte:

$$p = 0.99575, r = 1.0, f = 0.99787 \tag{5.4}$$

Es konnten also alle Substantive aus der Trainingsmenge klassifiziert werden, wenige wurden falsch klassifiziert. Um festzustellen, wie gut die Trainingsmenge ist, wurde diese manuell überprüft. Es wurden 200 Substantive willkürlich ausgewählt und festgestellt, dass diese korrekt eingeteilt sind.

5.2.2 *Wortartbestimmung*

Ebenso wie die Ermittlung des Geschlechts der Substantive basiert das Verfahren zur Bestimmung der Wortart auf einem Compact Patricia Trie.

Wird für die Trainingsmenge (Auschnitt siehe Abschnitt A.2 auf Seite 84) die Wortart ermittelt, ergeben sich folgende Werte:

$$p = 0.98930, r = 0.99968, f = 0.99447 \tag{5.5}$$

Weder sind alle der klassifizierten Grundformen korrekt klassifiziert, noch kann das Verfahren für jede Grundform eine Wortart bestimmen. Eine manuelle Auswertung von 200 Grundformen, die willkürlich aus der Trainingsmenge ausgewählt wurden, ergab, dass von diesen 8 falsch klassifiziert sind.

5.2.3 *Silbentrennung*

Der für die Silbentrennung eingesetzte Algorithmus von Liang (*Liangs Algorithmus*) basiert auf einer Liste von Trennmustern. Die Trennmuster werden auf Basis einer Wortliste erstellt, die in Silben getrennte Wörter enthält.

Als Referenz wurde eben jene Wortliste verwendet. Verglichen wurde jeweils, ob das Wort der Wortliste genau so getrennt wurde, wie es in der Wortliste getrennt steht. Es wurde nicht gezählt, wie viele falsche Silben oder Trennstellen ein Wort enthielt, sondern lediglich, ob das Wort korrekt getrennt wurde oder nicht.

Da das Programm `patgen` zum Erstellen der Trennmuster so konfiguriert wurde, dass sämtliche in der Trainingsmenge enthaltenen Wörter korrekt getrennt werden, ergeben sich die Werte

$$p = 1.0, r = 1.0, f = 1.0 \tag{5.6}$$

Wichtig ist jedoch, dass Wörter mit weniger als 4 Zeichen nicht getrennt werden, auch findet sich keine Trennstelle in den ersten bzw. letzten beiden Buchstaben eines Wortes.

Bei der Auswertung von 200 willkürlich ausgewählten Wörtern der Trainingsmenge wurden 10 falsch getrennte Wörter gezählt. Hierunter befinden sich allerdings 8, bei denen fälschlicherweise der erste bzw. der letzte Buchstabe nicht abgetrennt wurde, was zwar den Regeln der Rechtschreibung entspricht (vgl. [fdS06, S. 103]), jedoch nicht der phonologischen Einteilung in Silben.

5.2.4 *Wurzel und Präfix des Wortes*

Für das Abtrennen der Wurzel bzw. des Präfixes von einem Wort wird das Verfahren von [BW05] eingesetzt, welches auch näher in Abschnitt 4.1.1 auf Seite 40 beschrieben ist.

Grundlage für die Listen, mit denen die Compact Patricia Trees trainiert wurden, sind Wörter, die in Suffixe, Präfixe und Wortstämme geteilt wurden (für einen Auszug aus den Trainingsdaten siehe Abschnitt A.3 auf Seite 85). Jene Wörter wurden als Referenz für die Evaluation der Abtrennung der Wurzel genutzt. Es wurde verglichen, ob der am weitesten rechts stehende Teil eines Wortes mit dem übereinstimmt, das das in FLEXI eingesetzte Verfahren als Wurzel zurückliefert. Es ergaben sich folgende Werte:

$$p = 0.90525, r = 1.0, f = 0.95027 \tag{5.7}$$

Der hohe Recall ergibt sich daraus, dass für jedes Wort eine Wurzel zurückgeliefert werden kann. Eine manuelle Auswertung von 200 willkürlich ausgewählten Wörtern aus der Menge der Wörter, die in Präfixe, Suffixe und Stämme geteilt sind, ergab, dass bei 33 Wörter nicht korrekt getrennt wurden. Diese recht hohe Fehlerquote resultiert vor allem daraus, dass auch in lateinisch-, englisch- und französisch stämmige Morpheme zerlegt wurde, und dies bei Wörtern wie z.B. *Internet* in die Zerlegung *Int-ern-et*, wobei

Int als deutscher Stamm, *ern* und *et* als lateinische Suffixe gekennzeichnet wurden.

Da die Abtrennung des Präfixes besonders für Verben von Bedeutung ist, wurden 200 willkürlich ausgewählte komplexe Verben manuell daraufhin untersucht, ob deren Präfix korrekt erkannt wird. Bei 6 Verben wurde der Präfix nicht korrekt abgetrennt.

5.3 EVALUATION VON FLEXI

Nachdem die einzelnen Verfahren evaluiert wurden, werden im Folgenden die Untersuchungen zu der Güte von FLEXI beschrieben. Die ursprüngliche Trainingsmenge wurde stark gesäubert und somit reduziert:

- Es wurden zu lange Grundformen gelöscht.[3]
- Es wurden niederfrequente Substantive gelöscht.[4]
- Nach Wortarten einzeln getrennt, gab es zwar Probleme bei der Performanz, jedoch war die Ausführung von FLEXI möglich. Unmöglich war die Ausführung mit allen Daten[5], da durch die hohe Anzahl der Klassifizierungen das Erstellen des Klassifizierers zu viele Ressourcen benötigt wurden. Somit wurden die Grundformen entfernt, die bei optimaler Einstellung[6] falsch klassifiziert wurden.
- Da nicht alle enthaltenen Grundformen vollständig flektiert waren, wurden die Grundformen gelöscht, zu denen es für bestimmte Merkmalskombinationen keine Form gab.[7]

Grundsätzlich wird bei der Evaluierung wie folgt vorgegangen: Die in dem verwendeten Lexikon enthaltenen

3 Eine manuelle Analyse hat ergeben, dass z.B. Substantive mit mehr als 15 Buchstaben Komposita sind.

4 Da gerade die Substantive überwiegend vorhanden waren, wurden alle Substantive gelöscht, die im Nachrichten-Korpus von 2008 des Wortschatz-Projektes der Universität Leipzig (zu finden online unter `http://wortschatz.uni-leipzig.de/`) mit einer Frequenz unter 150 vorkamen

5 In der verwendeten Version 1.6 von Java kann der virtuellen Maschine nicht der gesamte zur Verfügung stehende Speicher zugeteilt werden, sondern nur ein fester maximaler Wert.

6 Zu den unterschiedlich getesteten Einstellung in Abschnitt 5.3.3 auf Seite 71 mehr.

7 Für einige Substantive existierten z.B. nur Nominativ, Dativ und Akkusativ Singular.

Grundformen werden von FLEXI flektiert und als eine Liste von Tripeln (Grundform, flektierte Form, Merkmalsbezeichner) ausgegeben. Diese werden dann mit der Liste von Tripeln verglichen, die die Grundlage für das Lexikon war.

5.3.1 *Zählweisen*

Aus den unterschiedlichen Anwendungsfällen, die mit den Ergebnissen von FLEXI möglich sind, ergeben sich unterschiedliche Zählweisen. Die generierten Listen von Tripeln (bestehend aus Grundform, flektierter Form und Bezeichner der morphosyntaktischen Merkmalskombination) werden dreifach evaluiert:

ZÄHLUNG DER WORTFORMEN Es werden lediglich die flektierten Wortformen gegen die Referenzliste verglichen.

ZÄHLUNG DER TRIPEL Bei dem Vergleich mit der Referenzliste werden die gesamten Tripel verglichen.

ZÄHLUNG DER WÖRTER Bei dieser Art zu evaluieren, wird untersucht, ob für eine Grundform alle flektierten Formen mit ihren Bezeichnern für die morphosyntaktischen Merkmalskombinationen korrekt generiert wurden.

Evaluation am Beispiel

Als Beispiel dient ebenso wie in Kapitel 4 das Wort *oft*. Referenzdaten sind die Tripel (*oft*, *oft*, *Pos*), (*oft*, *öfter*, *Komp*) und (*oft*, *oftest*, *Sup*). Angeommen FLEXI erzeugt bei der Flexion von *oft* folgende Tripel: (*oft*, *oft*, *Pos*), (*oft*, *öfter*, *Sup*), (*oft*, *oftest*, *Sup*) und (*oft*, *öftest*, *Sup*). Es ergeben sich folgende Werte:

ZÄHLUNG DER WORTFORMEN

$$p = \frac{3}{4}, r = \frac{3}{3} = 1, f = \frac{2 * \frac{3}{4} * 1}{\frac{3}{4} + 1} = \frac{6}{7} \tag{5.8}$$

ZÄHLUNG DER TRIPEL

$$p = \frac{2}{4} = \frac{1}{2}, r = \frac{2}{3}, f = \frac{2 * \frac{2}{4} * \frac{2}{3}}{\frac{2}{4} + \frac{2}{3}} = \frac{4}{7} \tag{5.9}$$

ZÄHLUNG DER WÖRTER

$$p = \frac{0}{1} = 0, r = \frac{0}{1} = 0, p = r = 0 \rightarrow f = 0 \qquad (5.10)$$

5.3.2 *Vergleichswert als Grundlage*

Für Vergleiche mit späteren Messungen soll zunächst eine Grundlage geschaffen werden. Hierbei soll die Frage geklärt werden: Was kann mit einfachsten Mitteln erreicht werden? Es wird das Lexikon wie gewohnt eingelesen, allerdings wird für die Klassifikation ein Entscheidungsbaum verwendet, der auf 0 Instanzen der Trainingsdaten basiert und als einzige Klassifikation die häufigste Klassifikation der Trainingsdaten zurückliefert. Es handelt sich also um einen Entscheidungsbaum mit genau einem Blatt. Die Zeichen der Grundformen werden lediglich von rechts nach links markiert und es können somit keine Morphemgrenzen oder der Stammvokal verwendet werden.

Die flektierten Formen basieren auf ca. 23400 Substantiven, ca. 8800 Adjektiven, ca. 13000 Verben und ca. 44700 Eigennamen. Für den gesamten Datenbestand wurde eine spezielle Filterung vorgenommen: Es wurden nur die Grundformen verwendet, die bei der nach Wortarten unterteilten Klassifizierung korrekt klassifiziert wurden und es sind ca. 56000 Grundformen enthalten. Dies war notwendig, da der Datenbestand sonst zu groß für die Klassifizierer geworden wäre und somit FLEXI nicht ausführbar gewesen wäre.

Die Tabelle 5.2 enthält die Auswertung zunächst nach Wortarten unterschieden und dann für den gesamten Datenbestand.

BEWERTUNG DER ERGEBNISSE Besonders gute Ergebnisse waren nicht zu erwarten. Unterscheidet man die Ergebnisse nach der Zählweise, lässt sich zusammengefasst folgendes hervorheben:

- Betrachtet man nur die Wortformen, sticht besonders die hohe Precision bei allen vier Wortarten hervor, die allerdings in jedem Fall mit einem niedrigen Recall einhergeht.
- Besonders gut schneiden bei den Tripeln eigentlich nur die Eigennamen ab, da hier die wenigsten Unre-

	Zählweise	Precision	Recall	F-Wert
Eigennamen	Wortformen	0,95067	1,00000	0,97471
	Tripel	0,97496	0,95038	0,96251
	Wörter	0,88139	0,88094	0,88117
Verben	Wortformen	0,71615	0,42293	0,53180
	Tripel	0,66779	0,43820	0,52916
	Wörter	0,00092	0,00092	0,00092
Substantive	Wortformen	0,99914	0,35467	0,52351
	Tripel	0,50813	0,51670	0,51238
	Wörter	0,01354	0,01353	0,01354
Adjektive	Wortformen	1,00000	0,10858	0,19590
	Tripel	0,13095	0,07405	0,09460
	Wörter	0,11799	0,11798	0,11799
Alle Daten	Wortformen	0,15914	0,15999	0,15957
	Tripel	0,13946	0,13946	0,13946
	Wörter	0,14430	0,14409	0,14419

TABELLE 5.2: Grundlage der Evaluation für den Vergleich

gelmäßigkeiten zu beobachten sind und es die wenigsten Klassifikationen gibt.

- Die recht strenge Zählweise der Wörter ergibt durchweg nur sehr schlechte Werte. Besonders schlecht werden die Verben gebeugt, da hier die größten Unregelmäßigkeiten vorkommen.

Für den gesamten Datenbestand ist diese Messung recht ungünstig, da die häufigste Klassifizierung eine der Eigennamen ist und somit die Verben, Adjektive und Substantive von Grund auf falsch flektiert werden.

5.3.3 *Evaluation nach Wortarten*

Zunächst wurde nach Wortarten unterschieden evaluiert. Hierzu wurden alle möglichen Merkmalsermittler verwendet und zudem die Morphemgrenzen und die Stammvokale gekennzeichnet. Dies ist die umfangreichste Konfigurationsmöglichkeit.

Naiver Bayes'scher Klassifizierer

Die Tabelle 5.3 enthält die Ergebnisse für den Naiven Bayes'schen Klassifizierer.

	Zählweise	Precision	Recall	F-Wert
Eigennamen	Wortformen	0,99074	0,99032	0,99053
	Tripel	0,99403	0,97410	0,98397
	Wörter	0,96300	0,96317	0,96308
Verben	Wortformen	0,87524	0,69921	0,77738
	Tripel	0,83804	0,72144	0,77538
	Wörter	0,48172	0,48191	0,48182
Substantive	Wortformen	0,94251	0,94198	0,94225
	Tripel	0,92893	0,92707	0,92800
	Wörter	0,77826	0,77823	0,77824
Adjektive	Wortformen	0,87501	0,87349	0,87425
	Tripel	0,81786	0,81786	0,81786
	Wörter	0,81782	0,81783	0,81783

TABELLE 5.3: Evaluation nach Wortarten mit dem Naiven Bayes'schen Klassifizierer

BEWERTUNG DER ERGEBNISSE Insgesamt wurden die Erwartungen erfüllt, da die Werte recht gut sind. Hervorstechend sind die Verben, die durch viele Unregelmäßigkeiten recht schlecht klassifiziert werden konnten. Dies ist nach einer genaueren Untersuchung leicht zu erklären: Ein unregelmäßiges Verben wie *gehen* bildet eine eigene Klassifizierung, da kein Wort aus Sicht von FLEXI genau so flektiert. Das davon abgeleitete Verb *weggehen* hat wiederum auch eine eigene Klassifizierung, da sich hier z.B. bei der 1. Person Singular Präsenz Indikativ das *weg* verschieben kann (vgl. *ich gehe weg*). Dadurch entstanden insgesamt sehr viele Klassifizierungen (bei den Verben sind das ca. 4200, bei den Eigennamen als Vergleich nur 10) für die der Naive Bayes'sche Klassifizierer nicht so gut geeignet ist.

Verbesserung der Ergebnisse

Es wurde auf verschiedene Arten versucht, die in Tabelle 5.3 gezeigten Ergebnisse zu verbessern.

NACHSCHLAGEN IM LEXIKON Man kann bei der Benutzung von FLEXI die Option aktivieren, so dass falls ein eingegebenes Wort im Lexikon enthalten ist, die Daten aus dem Lexikon zurückgegeben werden. Dies bringt eine hohe Qualität für alle Grundformen aus dem Lexikon, da FLEXI erst zu raten beginnt, wenn ein Wort nicht im Lexikon

steht. Dadurch steigen Precision, Recall und F-Wert in allen Fällen auf 1.0.

GEWICHTEN DER MERKMALE Da der Naive Bayes'sche Klassifizierer die Möglichkeit bietet, einzelne Merkmale zu gewichten, wurde diese Möglichkeit untersucht. Getestet wurde, indem 100 mal für die Merkmale zufällige Gewichtungen zwischen 0.1 und 100 vergeben wurden, was jedoch keinerlei Verbesserung oder Verschlechterung mit sich brachte.

MORPHEMKENNZEICHNUNGEN WEGLASSEN Eventuell werden durch das Kennzeichnen der Morphemgrenzen und des Stammvokals zwar weniger Klassifikationen erzeugt, es ist aber auch durchaus möglich, dass durch hierbei entstandene Fehler die Qualität insgesamt gesteigert werden kann, wenn die Zeichen lediglich von rechts nach links durchnummeriert werden. Es wurde jedoch festgestellt, dass die Ergebnisse nur teilweise leicht besser wurden, bei den Verben sogar deutlich schlechter. Siehe hierzu Tabelle 5.4 auf Seite 72, die nur die Wortarten enthält, bei denen sich die Ergebnisse deutlich verbessert haben.

	Zählweise	Precision	Recall	F-Wert
Eigennamen	Wortformen	0,99919	0,99976	0,99947
	Tripel	0,99847	0,97892	0,98860
	Wörter	0,98071	0,98062	0,98066
Substantive	Wortformen	0,94251	0,94198	0,94225
	Tripel	0,92893	0,92707	0,92800
	Wörter	0,77826	0,77823	0,77824
Adjektive	Wortformen	0,89539	0,93859	0,91648
	Tripel	0,88366	0,93180	0,90709
	Wörter	0,79240	0,79240	0,79240

TABELLE 5.4: Verbesserte Evaluation nach Wortarten mit dem Naiven Bayes'schen Klassifizierer

Entscheidungsbaum

Im Folgenden wird nun auf den gleichen Daten mit dem Entscheidungsbaum evaluiert, und so festgestellt, ob es dadurch zu Verbesserungen oder Verschlechterungen kommt.

Den Entscheidungsbaum mit den gleichen Daten wie den Naiven Bayes'schen Klassifizierer zu trainieren, war nur bei den Adjektiven möglich. Bei den restlichen Wortarten scheiterte es an der Komplexität. Entweder enthalten die Trainingsdaten zu viele Instanzen oder zu viele Klassifizierungen.

Daher wurden die Trainingsdaten gefiltert: Die Substantive auf 1 %, die Eigennamen auf 10 % und die Verben sogar auf 0.1 %. Bei der Reduzierung blieb die Verteilung der Klassifizierungen erhalten. Somit ergeben sie für den Entscheidungsbaum die Werte in Tabelle 5.5 auf Seite 73. Als Referenzdaten wurden die gleichen Daten verwendet wie zur Evaluierung des Naiven Bayes'schen Klassifizierers.

	Zählweise	Precision	Recall	F-Wert
Eigennamen	Wortformen	0,98020	0,98110	0,98065
	Tripel	0,60055	0,95661	0,73787
	Wörter	0,46123	0,46239	0,46181
Verben	Wortformen	0,51318	0,44153	0,47466
	Tripel	0,36692	0,40610	0,38552
	Wörter	0,00131	0,00131	0,00131
Substantive	Wortformen	0,94251	0,94198	0,94225
	Tripel	0,50333	0,54132	0,52164
	Wörter	0,02900	0,02900	0,02900
Adjektive	Wortformen	0,99969	0,99938	0,99954
	Tripel	0,99960	0,99935	0,99947
	Wörter	0,99909	0,99909	0,99909

TABELLE 5.5: Evaluation nach Wortarten mit dem Entscheidungsbaum (C4.5)

Es ist bei allen Wortarten, deren Trainingsdaten reduziert wurden, eine Verschlechterung festzustellen. Einzig bei den Adjektiven wurde ein sehr gutes Ergebnis erzielt, vor allem deswegen, weil der Entscheidungsbaum die Eigenschaft hat, die Trainingsmenge abzubilden, insofern darin keine Inkonsistenzen enthalten sind. Herausragend schlecht schneiden die Verben ab, bei denen allerdings auch am meisten reduziert wurde.

Da der verwendete Algorithmus zum Erstellen des Entscheidungsbaumes zwar recht fortschrittlich ist, aber auch sehr komplex, wurde alternativ der ID3-Algorithmus verwendet. Dieser ist der Vorgänger des C4.5-Algorithmus und hat den Vorteil, dass er sehr einfach aufgebaut ist.

Hierfür mussten die Verben und Adjektive gar nicht, die Substantive auf 75 % die Eigennamen auf 10 % reduziert werden. Es wurden die Ergebnisse aus Tabelle 5.6 auf Seite 74 erzielt.

	Zählweise	Precision	Recall	F-Wert
Eigennamen	Wortformen	0,95435	0,96607	0,96018
	Tripel	0,96214	0,95128	0,95668
	Wörter	0,83697	0,83783	0,83740
Verben	Wortformen	1,00000	1,00000	1,00000
	Tripel	1,00000	1,00000	1,00000
	Wörter	1,00000	1,00000	1,00000
Substantive	Wortformen	1,00000	0,47873	0,64749
	Tripel	0,58393	0,60672	0,59511
	Wörter	0,14206	0,14203	0,14204
Adjektive	Wortformen	0,99969	0,99938	0,99954
	Tripel	0,99960	0,99935	0,99947
	Wörter	0,99909	0,99909	0,99909

TABELLE 5.6: Evaluation nach Wortarten mit dem Entscheidungsbaum (ID3)

Der Vergleich zwischen C4.5 und ID3 zeigt: Dadurch, dass bei letzterem in weniger Fällen weniger reduziert werden musste, schneidet er besser ab. Offensichtlich kann er besser mit größeren Mengen von Klassifizierungen umgehen.

Der ID3-Algorithmus schneidet in den Fällen, in denen nicht reduziert wurde, auch deutlich besser als der Naive Bayes'sche Klassifizierer ab (vgl. Tabelle 5.3 auf Seite 71 mit Tabelle 5.6 auf Seite 74).

Es konnten durchweg nur Verschlechterungen der Werte festgestellt werden, wenn auf die Kennzeichnung der Morphemgrenzen sowie des Stammvokals verzichtet wurde.

Fazit und Vergleich

Bei den bisherigen Auswertungen sind die Vor- und Nachteile der einzelnen Klassifizierer deutlich geworden: der Naive Bayes'sche Klassifizerer ist gut geeignet für größere Datenmengen, aber klassifiziert dafür insgesamt nicht besonders gut. Mit dem Entscheidungsbaum lassen sich nicht besonders große Datenmengen handhaben. Wenn die Trainingsdaten aber wenig umfangreich sind, erzielt er deut-

lich bessere Ergebnisse. Im Vergleich zu den Ausgangswerten in Tabelle 5.2 auf Seite 70 haben beide Klassifizierer deutlich besser abgeschnitten, insofern die Trainingsdaten nicht allzu sehr reduziert wurden (wie z.B. bei den Eigennamen mit dem C4.5-Algorithmus, vgl. Tabelle 5.5 auf Seite 73).

5.3.4 *Evaluation der gesamten Daten*

Das Hauptaugenmerk von FLEXI liegt nunmehr nicht auf einer wortartabhängigen Flexion von Grundformen, sondern auf der Flexion von einer beliebigen (flektierbaren) Grundform. Außerdem sollen verschiedene Konfigurationen untersucht werden, um Merkmale zu identifizieren, die eventuell einen negativen Einfluss auf die Ergebnisse haben.

Mit den verschiedenen Klassifizierern wurden die Ergebnisse in Tabelle 5.7 auf Seite 75 erzielt. Für den Entscheidungsbaum mussten die Trainingsdaten bei dem C4.5-Algorithmus auf 0,5 %, bei dem ID3-Algorithmus auf 15 % der Daten reduziert werden. Wie bei den bisherigen Reduzierungen blieb die Verteilung der Klassifizierungen erhalten.

	Zählweise	Precision	Recall	F-Wert
Bayes	Wortformen	0,98211	0,94577	0,96360
	Tripel	0,97732	0,95143	0,96420
	Wörter	0,95299	0,95289	0,95294
C4.5	Wortformen	0,59280	0,54035	0,56536
	Tripel	0,06687	0,25429	0,10590
	Wörter	0,00853	0,00853	0,00853
ID3	Wortformen	0,90388	0,59534	0,71786
	Tripel	0,88025	0,24474	0,38300
	Wörter	0,70288	0,70295	0,70291

TABELLE 5.7: Evaluation aller Daten

Wiederum konnte der C4.5-Algorithmus nur mit weniger Daten trainiert werden als der ID3-Algorithmus. Der Naive Bayes'sche Klassifizierer erzielte jedoch deutlich bessere Ergebnisse und kam ohne eine Reduzierung aus.[8]

8 Reduzierte man die Daten auf 5 % und verwendete den ID3-Algorithmus, war dieser deutlich besser als der C4.5-Algorithmus mit der gleichen Reduzierung.

Da die beiden Entscheidungsbaum-Algorithmen jedoch deutlich schlechter abschnitten als der Naive Bayes'sche Klassifizierer, wird im Folgenden nur noch auf diesen detaillierter eingegangen.

Verbesserung der Ergebnisse

Auch bei der Evaluation aller Daten wurden Untersuchungen angestellt, wie sich die Ergebnisse verbessern lassen.

NACHSCHLAGEN IM LEXIKON Aktiviert man die Option von FLEXI, bei einer eingegeben Grundform zunächst im Lexikon nachzuschlagen und bei einem Fund die dort gespeicherten Daten zurückzugeben, erhält man auch hier Precision, Recall und F-Wert gleich 1, da als Referenzdaten die Daten verwendet werden, auf denen das Lexikon basiert.

WEGLASSEN DER MORPHEMKENNZEICHNUNG Wie schon bei der Evaluation getrennt nach Wortarten kann es hilfreich sein, die Morphemkennzeichnungen und die Kennzeichnung des Stammvokals wegzulassen. Allerdings ergaben sich hierbei leichte Verschlechterungen.

WEGLASSEN EINIGER MERKMALE Da es möglich ist, dass einige der in Kapitel 2 ermittelte Merkmale doch nicht für die Flexion bestimmend sind, wurden verschiedene Konfigurationen untersucht, bei der jeweils ein Merkmal weggelassen wurde. Die Messungen ergaben jedoch, dass sich die Ergebnisse entweder gar nicht oder leicht verschlechtert haben.

Härtetest

Wie schon zu Beginn von Abschnitt 5.3 beschrieben, bestehen die gesamten Daten nicht aus der Summe der Daten der einzelnen Wortarten. Es wurden also diejenigen Grundformen aus der Trainingsmenge herausgenommen, die bei der Flexion der einzelnen Wortarten mit dem Naiven Bayes'schen Klassifizierer falsch flektiert wurden. Nimmt man nun diese reduzierten Daten als Trainingsmenge, die nicht-reduzierten Daten als Referenzdaten, ergeben sich die Werte in Tabelle 5.8 auf Seite 77.

Nun kann dieses Ergebnis dadurch verbessert werden, indem bei einer Grundform, die im Lexikon steht, einfach

deren zugehörige Daten aus dem Lexikon zurück gegeben werden. Auf diese Art sind folgende Ergebnisse die Werte in Tabelle 5.8 auf Seite 77 zu erzielen.

	Zählweise	Precision	Recall	F-Wert
Ohne Nachschlagen	Wortformen	0,76628	0,65539	0,70651
	Tripel	0,63531	0,71039	0,67076
	Wörter	0,55738	0,55723	0,55730
Mit Nachschlagen	Wortformen	0,77631	0,67374	0,72140
	Tripel	0,64867	0,73350	0,68848
	Wörter	0,58386	0,58374	0,58380

TABELLE 5.8: Härtetest für FLEXI

Die reduzierten Daten enthielten ca. 49000, die nicht-reduzierten Daten ca. 88000 Grundformen.

Fazit

Nachdem die Evaluation der Eigennamen gezeigt hat, dass die Flexion von sehr regelmäßig flektierenden Wortarten wie den Eigennamen mit FLEXI sehr gut und von Wortarten, deren Veränderungen recht komplex sind (wie die der Verben) eher mittelmäßig gut funktioniert, beherrscht FLEXI auch die Flexion der gesamten Daten recht gut. Ob sich dies jedoch für die Anwendung eignet, soll der Abschnitt 5.3.5 zeigen.

5.3.5 *Evaluation eines Anwendungsfalls*

Wie schon in Kapitel 1 erwähnt wurde, ist ein denkbarer Anwendungsfall für FLEXI eine vollständige Wortliste zu erzeugen.[9] Da so jedoch keinerlei weitere Angaben über die vorhandenen Wörter möglich sind, soll im Folgenden

9 Weitere denkbare Anwendungsmöglichkeiten sind vielfältig, insbesondere mit der Kernkomponente von FLEXI Wortveränderungen zu erfassen. Es können z.B. Analysen auf Zeitreihen von Texten unterschiedlichen Alters ergeben, wie sich Wörter über die Zeit hinweg verändern. So könnten dann eventuell Aussagen darüber getroffen werden, wie Wörter einer späteren Zeitperiode in einer früheren ausgesehen habe mögen. Ein gänzlich anderer denkbarer Anwendungsfall ist die Beobachtung von Derivationsprozessen, also z.B. zu erforschen, wie sich Adjektiv-Stämme verändern, wenn sie substantiviert werden.

eine Wortliste auf Fehler überprüft werden, deren Einträge Angaben zur Frequenz[10] enthalten.

Hierfür wurde eine Wortliste genutzt, die ca. 470000 flektierte Wortformen beinhaltet.[11] Jedes darin enthaltene Wort wurde mit dem Verfahren, das in Abschnitt 4.1.1 auf Seite 40 beschrieben wurde, auf seine Grundform reduziert und diese von FLEXI gebeugt. Ist die ursprüngliche Form nicht in den von FLEXI erzeugten Wortformen enthalten, wird sie als fehlerhaft markiert.

Diese Zählung ergab, dass sich in der Wortliste ca. 96000 falsche Wortformen befanden. Eine manuelle Auswertung von 200 willkürlich ausgewählten Wortformen hatte zum Ergebnis, dass 181 fälschlicherweise als falsch markiert wurden. Einen hohen Anteil dieser 181 Worte machen Komposita aus, die FLEXI offensichtlich Probleme bereiten.

5.4 FAZIT

Es folgt ein kurzes Fazit, in dem reflektiert werden soll, ob die in Kapitel 1 beschriebenen Ziele eingehalten wurden.

5.4.1 *Flexion unbekannter Wörter*

Es wurde sehr ausführlich beschrieben, welche Merkmale für die Flexion bestimmend sind. Jedoch lassen sich nicht alle Merkmale bei unbekannten Wörtern korrekt bestimmen (wie z.B. die ursprüngliche Herkunft). Auch bereiten die vielen Ausnahmen und unregelmäßig flektierten Wortformen die größten Probleme.

Entscheidend ist zudem die Komplexität der Daten. Wurde mit einem Entscheidungsbaum ohne Reduzierung der Trainingsdaten klassifiziert, wurden sehr gute Ergebnisse erzielt. Oft jedoch waren die Daten zu komplex für die Verwendung des Entscheidungsbaums ohne eine Reduzierung der Daten, so dass der Naive Bayes'sche Klassifizierer eingesetzt werden musste. Dieser wiederum eignet sich eher für geringere Anzahl an Klassifizierungen.

Folgende Veränderungen könnten daher die Ergebnisse verbessern:

10 Gemessen wurde die Häufigkeit des Vorkommens im jeweiligen Korpus.

11 Die ursprüngliche Wortliste enthielt 4800000 flektierte Wortformen. Sie wurde jedoch durch eine repräsentative Auswahl auf eine handlichere Größe reduziert.

- Die Vereinfachung der Sprache könnte helfen, die Komplexität selbiger zu reduzieren. Dies ist jedoch ein Prozess, der viel Zeit in Anspruch nehmen würde und nicht gesteuert werden kann.
- Entwicklung eines Algorithmus zum Erstellen des Entscheidungsbaumes, der mit sehr komplexen Daten umgehen kann.
- Die Erweiterung der Speicher- und Rechenressourcen, um die Berechnung des Entscheidungsbaumes mit bisherigen Algorithmen zu ermöglichen.
- Eine Reduzierung der Klassifizierungen ist zudem möglich, wenn die Umlautung als solche erkannt werden könnte. Bisher wird die Umlautung im optimalen Fall durch das Ersetzen des Stammvokals mit *ö* bzw. *ü* bzw. *ä* beschrieben. Eine Verbesserung wäre, wenn im Falle der Umlautung der Stammvokal als „umlautend“ markiert werden würde, dies ist jedoch sehr schwer möglich, da FLEXI darauf ausgelegt sein müsste, phonologische Prozesse anhand der Veränderung auf Buchstabenebene zu erkennen.
- Bei den Verben hat sich eine Schwachstelle von FLEXI gezeigt: Bei zusammengesetzten Verben (wie z.B. *weggehen*) kann sich der Präfix verschieben (wie z.B. bei *ich gehe weg*). Dieses Verschieben wird von dem verwendeten Algorithmus *diff* nicht erkannt, da die zugrunde liegende LCS nicht geeignet ist. Ein alternativer Algorithmus um Veränderungen festzustellen ist also zu finden.[12]

5.4.2 *Sprachunabhängigkeit*

Die Sprachunabhängigkeit konnte konsequent beachtet werden. Dies betrifft zunächst das Lexikon: Jeder Eintrag besteht aus Merkmalen, flektierten Formen, Kennzeichnungen der Buchstaben und einer Grundform. So kann jede flektierende Sprache erfasst werden, denn in jeder flektierenden Sprache

- lässt sich eine Grundform bestimmen.

12 Eine Neuentwicklung von *diff* würde den Rahmen dieser Arbeit sprengen; auch widerspräche es der Tradition der *Automatischen Sprachverarbeitung*, Probleme der Linguistik mit Methoden der Informatik zu lösen, da hierfür ein spezielles, der morphologischen Veränderung angepasstes *diff* zu entwickeln wäre.

- lassen sich logischerweise flektierte Formen bilden.
- werden Veränderungen an bestimmten Stellen vollzogen, die sich mehr oder weniger gut automatisch erfassen lassen.
- sind Merkmale für die Flexion bestimmend. Je nachdem wie gut sich die Merkmale automatisiert bestimmen lassen und wie wenig Ausnahmen existieren, desto besser wird FLEXI funktionieren.

Auch FLEXI selbst ist so allgemein gehalten, dass es auf andere Sprachen angewandt werden kann. Es wurde zwar vom Deutschen ausgehend entwickelt, aber keinerlei Eigenheiten des Deutschen beachtet.

APPENDIX

A

TRAININGSDATEN

In diesem Abschnitt des Anhangs befinden sich Auszüge aus den Trainingsdateien. Diese sind willkürlich gewählt, erheben keinerlei Anspruch darauf repräsentativ zu sein und können Fehler enthalten.

A.1 GESCHLECHT DER SUBSTANTIVE

Die Klassifizierung *N* bzw. *M* bzw. *F* steht für *Neutrum* bzw. *Maskulinum* bzw. *Femininum*.

Bremsenwerk	N
Fedelhörn	N
Helmkraut	N
Lachend	N
Ortscheit	N
Scherzgeschäft	N
Theriakbüchslein	N
Zeitschriftenexperiment	N
Aufstiegskandidat	M
Bordell-Betreiber	M
Dreiländergipfel	M
Fehlschluss	M
Gesamtverbrecher	M
Hohehorst	M
Kleinbauernverband	M
Lieblingsschauspieler	M
Müll-Tourismus	M
Plantisch	M
Risalit	M
Siegelstempel	M
Südost	M
Verdauungsvorgang	M
Wohngelaß	M
Altstadtsanierung	F

Batik	F
Bundesplazierung	F
Dysthyreose	F
Faktenauskunft	F
Gebirgstauglichkeit	F
Handelszentrale	F
Jaroshinskaya	F
Kraftdroschke	F
Lückenschlußmaßnahme	F
Nachbesserungsklausel	F
Pfahlbürgerschaft	F
Reduktionsprobe	F
Schneesicherheit	F
Spritzenweihe	F
Theaterarbeit	F
Verhütung	F
Widerstandswoche	F

TABELLE A.1: Auszug aus den Trainingsdateien zur Geschlechterbestimmung der Substantive

A.2 WORTARTBESTIMMUNG

Die Klassifizierungen sind wie folgt:

N Nomen

A Adjektiv

NE Eigenname (Named Entity)

Weitere Klassifizierungen, die in diesem Auszug nicht vorkommen, sind z.B. *DET* für Determinator, *NUM* für Ordinalzahl oder *KONJ* für Konjunktion.

Grundkosten	N
zusammenkauern	V
asphaltiert	A
Slayer	N
Friedenspilgerweg	N
Verringert	N
Leugnerin	N
Union-Verlag	N
Bewerbungskostenzuschuß	N

RRK-Trainer	N
Referenzbauwerk	N
Gates-Kolumne	N
Baumschnitt	N
Valjoux-Kaliber	N
Lohnsteueranspruch	N
Frauenstelle	N
Badegewässerbericht	N
Entdeckungsspielraum	N
Chemiefaser-Zellstoff	N
Tickerverschnitt	N
Abwesenheitsnachweis	N
Dissidentensprecher	N
Redenschwingen	N
Acetylcholinvergiftung	N
Bitstromübertragung	N
Tram-Haltestelle	N
Sockenfilz	N
Schafskäsesalat	N
Ausnahmecharakter	N
Vorschläger	N
Bauabnahmekommission	N
Auffahrtkeil	N
Plattenbaufestung	N
Rechtsbedarf	N
Lederbahre	N
Stabelektrode	N
Rindleder	N
Kindesalter	N
Durchbildung	N
Graditz	NE
Cheikh	NE

TABELLE A.2: Auszug aus den Trainingsdateien zur Wortartbestimmung

A.3 MORPHEM-ZERLEGUNG

Die Klassifizierung für die ersten beiden Spalten beschreiben die Trennstelle von vorne bzw. von hinten betrachtet, die Klassifizierung der dritten Spalte beschreibt die Regeln zur Grundformreduzierung, so bedeutet *5en*, dass 5

Zeichen gelöscht und *en* angefügt werden muss, um die Grundform zu erhalten.

Von vorne betrachtet		Von hinten betrachtet		Grundformreduzierung	
amphetaminfälle	10	skeletthand	4	Kernkraftfreund	0
aufgedunsensten	3	zestbe	2	Satzanschlußlage	0
baseballbezogene	4	startabsage	4	Strumaektomie	0
bewässerungsprogrammen	2	ostspanische	9	Engolme	1
bücherlandschaften	6	pinkle	6	Trekking-Reise	0
determinierungs	2	nichtbare	9	Förderungsprojekte	1
eiauf	2	vegetationsgebiete	5	Prüfkartensätze	4atz
episthemologisches	4	definitionsliste	5	Session-Beendigung	0
fängerabschnitts	6	marktungsvertrag	4	Bühnenübersetzung	0
fordelich	9	kerosinsteuerbefreiung	7	Lockerungsgymnastik	0
gaußfilter	4	bandunterhaltung	7	versäumtem	2
gewesenseins	2	sowohlalsauch	0	Lipiden	2
haklsteckens	6	lutherkritik	6	ausstiegen	5eigen
herumquälte	5	klassenauswahl	4	Anzeigenplazierungen	2
hygienediensten	7	butterrahm	4	Journalistenbesuchen	2
kältetodholt	5	quecksilberplomben	7	Nebenvorteilen	2
klumpungs	9	formunabhängigen	8	Klubkameradinnen	3
kroatenhoch	7	muschelkörbchen	8	geschiedeneren	2
leerkaufte	4	auftragswerken	6	Hauptschulabschlüßen	4uß
mahlnen	7	inspirationen	11	Tunnelvarianten	1
militärhirnen	7	fährleutegenossen	6	Mittagsrundschauen	2
nachschweißen	4	schmelzlöten	5	Bierseideln	1
oberle	2	tagsdin	3	Radwandern	0
pfeffergewächsen	7	koalitionsspekulation	11	Musikförderern	1
propagandadecke	10	sammelun	2	knalltet nieder	15niederknallen
regieegomanen	5	schutzkleider	7	Chipkartenkontroller	0
rügevor	4	warsteinerfaßbier	4	Garagenfenster	0
schienengüter	8	gemeindesteuerer	7	Adreßstands	1
seelensgutes	7	hansabrauer	6	Fußball-Führungskataloges	2
sorgungssteige	8	holophras	5	Klosterhornes	2
stäubungsmechanismen	9	richtsentschlusses	9	Benutzungswertes	2
sylvesterverrecker	9	demergebnis	6	Telemarkschwungs	1
tonerentsorgung	5	antinos	7	Wachsdeckels	1
umherstreifen	5	quantentransports	10	Wehrhäuschens	1

B

ERKLÄRUNG

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort Datum Unterschrift

LITERATURVERZEICHNIS

[Bre09] Brenner, Andreas: *Heilbronn: Andreas Brenner siegt mit Nupafeed's Carlos*. Website, 2009. Erreichbar online unter `http://www.sportpferde-brenner.com/de/news/story/?id=989`; besucht am 7. Oktober 2009.

[Bus90] Bussmann, Hadumod: *Lexikon der Sprachwissenschaft*. Alfred-Kröner-Verlag, 1990.

[BW05] Biemann, C. und F. Witschel: *Rigorous Dimensionality Reduction through Linguistically Motivated Feature Selection for Text Categorization*. Proceedings of NODALIDA, 2005.

[Die06] Diestel, Reinhard: *Graphentheorie – Elektronische Ausgabe 2006*. Springer-Verlag Heidelberg, 2006.

[Die09] Diening, Deike: *Ruh' im Wipfel*. Website, 2009. Erreichbar online unter `http://www.tagesspiegel.de/zeitung/Die-Dritte-Seite-Selbstmord;art705,2774239`; besucht am 8. Mai 2009.

[Dra04] Draxler, Christoph: *Sprachdatenbanken*. In: Carstensen, Kai Uwe, Christian Ebert, Cornelia Endriss, Susanne Jekat, Ralf Klabunde und Hagen Langer (Herausgeber): *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Elsevier, Spektrum Akademischer Verlag, 2 Auflage, 2004.

[DtH92] Domenic, Marc und Poius ten Hacken: *Word Manager: A System for Morphological Dictionaries*. Olms, 1992.

[fdS06] Sprache, Institut für deutsche: *Deutsche Rechtschreibung – Regeln und Wörterverzeichnis*. PDF, 2006. Erreichbar online unter `http://www.ids-mannheim.de/reform/regeln2006.pdf`; besucht am 29. Mai 2009.

[FN88] Finkler, Wolfgang und Günter Neumann: *MORPHIX - A Fast Realisation of a Classification-*

Based Approach to Morphology. Proceedings of 4th OFAI, 1988.

[Hal92] Hall, T. A.: *Syllable structure and syllable-related processes in German*. Niemeyer, Tübingen, 1992.

[Hau] Hauser, Prof. Roland: *Morphologie in JSLIM*. Website. Erreichbar online unter `http://www.linguistik.uni-erlangen.de/clue/de/forschung-projekte/jslim/download.html`; besucht am 18. Juli 2009.

[Hir75] Hirschberg, D. S.: *A linear space algorithm for computing maximal common subsequences*. Commun. ACM, 18(6):341–343, 1975.

[HM76] Hunt, J. W. und M. D. McIlroy: *An Algorithm for Differential File Comparison*. Technischer Bericht CSTR 41, Bell Laboratories, Murray Hill, NJ, 1976.

[HQW06] Heyer, Gerhard, Uwe Quasthoff und Thomas Wittig: *Text Mining – Wissensrohstoff Text*. w3L, 2006.

[KRSSW07] Kunkel-Razum, Dr. Kathrin, Dr. Werner Scholze-Stubenrecht und Dr. Matthias Wermke: *DUDEN – Deutsches Universalwörterbuch*. Bibliographisches Institut & F.A. Brockhaus AG, 2007.

[Lez96] Lezius, Dr. Wolfgang: *Morphologiesystem Morphy*. In: *Linguistische Verifikation. Dokumentation zur ersten Morpholympics 1994*, Seiten 25–35. Niemeyer, 1996.

[Lia83] Liang, Franklin Mark: *Word hy-phen-a-tion by com-put-er*. Doktorarbeit, Stanford, CA, USA, 1983.

[MA05] Mark Aronoff, Kirsten Fudeman: *What is Morphology?* Blackwell Publishing, 2005.

[Mar82] Marantz, Alec: *Re Reduplication*. In: *Linguistic Inquiry 13*, Seiten 435–382. MIT Press, 1982.

[Mor68] Morrison, Donald R.: *PATRICIA—Practical Algorithm To Retrieve Information Coded in Alphanumeric*. J. ACM, 15(4):514–534, 1968.

[Neu] Neumann, Günter: *Morphix - A Fast and Portable Morphological Component for Inflectional Languages*. Website. Erreichbar online unter `http://www.dfki.de/~neumann/morphix/morphix.html`; besucht am 19. Juli 2009.

[Nil98] Nilson, Nils J.: *Artificial Intelligence – A New Synthesis*. Morgan Kaufmann Publishers, Inc, 1998.

[Qua07] Quasthoff, Uwe: *Deutsches Neologismenwörterbuch*. Walter de Gruyter, 2007.

[Qui86] Quinlan, R.: *Induction of decision trees*. Maschine Learning, 1(1):81–106, 1986.

[Qui93] Quinlan, J. Ross: *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RN03] Russell, Stuart J. und Peter Norvig: *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[Rö06] Römer, Christine: *Morphologie der deutschen Sprache*. A. Francke, 2006.

[TV09] Thieroff, Rolf und Petra M. Vogel: *Flexion*. Universitätsverlag Winter Heidelberg, 2009.

[Wea49] Weaver, Warren: *Recent Contributions to the Mathematical Theory of Communication*. 1949.